Calhoun
Institutional Archive of the Naval Postgraduate School

Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1984

An amphibious ship-to-shore simulation for use on an IBM PC.

Ritacco, Steven M.

http://hdl.handle.net/10945/19124

Downloaded from NPS Archive: Calhoun

DUDLEY
KNOX
LIBRARY

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

AN AMPHIBIOUS SHIP-TO-SHORE SIMULATION FOR USE ON AN IBM PC

by

Steven M. Ritacco

September 1984

Thesis Advisor: J.L. Wayman

Approved for public release; distribution unlimited.

T222490

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) An Amphibious Ship-to-Shore Simulation for Use on an IBM PC		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(*s*) Steven Michael Ritacco		8. CONTRACT OR GRANT NUMBER(*s*)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE September 1984
		13. NUMBER OF PAGES 202
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Amphibious Movement Simulation

Ship-to-Shore Simulation

Simulation for IBM PC

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This thesis completed implementation of an amphibious ship-to-shore simulation (called SHIPSHOR) for use on an IBM Personal Computer. The investigation included a description of the physical system being modelled, an explanation of the logic used by the model, a validation section, sensitivity analysis, and a thorough documentation section.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

The model is functioning and producing credible output as exhibited in the validation chapter. It is capable of operating under a variety of conditions to produce results which illustrate the build-up ashore of personnel and fire-power versus time. Its main application, as suggested within, is for use as a decision aid to the commander in operational planning and to the staff officer in procurement planning.

To be effective, SHIPSHOR needs continuous validation and modification. Model building is an evolutionary process which should not cease until the usefulness of the model has expired.

Approved for public release; distribution unlimited.

SHIPSHOR - An Amphibious
Ship-to-Shore Simulation
for Use on an IBM PC

by

Steven M. Ritacco
Major, United States Marine Corps
B.S., United States Naval Acedemy, 1973

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1984

R5786
c.1

ABSTRACT

This thesis completed implementation of an amphibious ship-to-shore simulation (called SHIPSHOR) for use on an IBM Personnel Computer. The investigation included a description of the physical system being modelled, an explanation of the logic used by the model, a validation section, sensitivity analysis, and a thorough documentation section.

The model is functioning and producing credible output as exhibited in the validation chapter. It is capable of operating under a variety of conditions to produce results which illustrate the build-up ashore of personnel and firepower versus time. Its main application, as suggested within, is for use as a decision aid to the commander in operational planning and to the staff officer in procurement planning.

To be effective, SHIPSHOR needs continuous validation and modification. Model building is an evolutionary process which should not cease until the usefulness of the model has expired.

TABLE OF CONTENTS

I. BACKGROUND 12
  A. THE PROBLEM 12
  B. PURPOSE 15

II. SIMULATIONS 16

III. THE SYSTEM 22

IV. THE MODEL 30

V. VALIDATION AND VERIFICATION 37
  A. GENERAL 37
  B. VALIDATION 38
    1. Initial Phase: Simplified Input File . . . 38
    2. Phase Two: Comparison to Large Scale Simulation 39
  C. SENSITIVITY ANALYSIS 55
    1. General 55
    2. Graphics and Numerics 57

VI. MODEL DOCUMENTATION 83

VII. CONCLUSIONS 86
  A. SUMMARY 86
  B. FUTURE ENHANCEMENTS 87
  C. FUTURE APPLICATIONS 87

APPENDIX A: GLOSSARY OF ACRONYMS 89

APPENDIX B: DECISION-MAKER'S MANUAL 92
  A. TABLE OF CONTENTS 92
  B. INTRODUCTION 92

C. MODEL DESCRIPTION 93
1. Capabilities 93
2. Input/Output Classes 94
3. Assumptions and Limitations 97
D. DEVELOPMENT AND EXPERIMENTATION 99
1. Development History 99
2. Verification and Validation 99
E. CURRENT AND ADDITIONAL APPLICATIONS 100
1. Current Use 100
2. Additional Applications 101
F. REFERENCES 101

APPENDIX C: USER'S MANUAL 102
A. TABLE OF CONTENTS 102
B. INTRODUCTION 102
C. DESCRIPTION OF THE MODEL 103
1. Overview 103
2. Methodology 109
3. Assumptions and Limitations 115
D. MODEL INPUT DATA 117
1. General Description 117
2. Detailed Descriptions 120
3. Data Collection and Maintenance 123
E. MODEL OUTPUT DATA 126
1. General Description 126
2. Detailed Description 126
F. SAMPLE RUN 128
G. VARIABLE NAMES AND LOCATIONS 128

APPENDIX D: PROGRAMMER'S MANUAL 129
A. TABLE OF CONTENTS 129
B. INTRODUCTION 129
C. MODEL SPECIFICATION 130
1. Purpose 130

2. Host System 130
3. Processing requirements 130
4. Language 130
5. Capabilities 130
6. Limitations 130
D. MODEL DESCRIPTION 131
1. Overview 131
2. General logic flow 131
E. DESCRIPTION OF ROUTINES 133
1. Routine Name: SETUP 133
2. Routine Name: INITIAL 134
3. Routine Name: Wave time sort routines . 135
4. Routine Name: EXEC 137
5. Routine Name: HRLD/BRLD 138
6. Routine Name: SPOTS/WELDEK 139
7. Routine Name: HLNCH/BLNCH 141
8. Routine Name: HUNLD/BUNLD 143
9. Routine Name: HRTN/BRTN 145
10. Routine Name: HARIV/BARIV 147
11. Routine Name: RESTART/BESTART 150
12. Routine Name: STOCH/STOCL 152
13. Routine Name: STNSH/STNSL 152
14. Routine Name: STSH/STSL 153
15. Routine Name: HCKLD 154
16. Routine Name: HIODCK 155
17. Routine Name: WSPOT 155
18. Routine Name: HALTSH/HALTSL 156
F. DATA BASE DESCRIPTION 157
1. File name: Landing Plan data file . . . 157
2. File name: T-1.DAT 159
3. File Name: Interactive/line input . . . 161
G. SOURCE LISTING 162
H. GLOSSARY OF VARIABLES 163

I. MODEL TEST RESULTS 163

APPENDIX E: 'HELP' FILE 165

A. TABLE OF CONTENTS 165

B. GENERAL 165

C. LANDING PLAN FILE 167

D. T-1.DAT FILE 168

E. INPUT PARAMETERS 169

F. SUBROUTINES 172

G. VARIABLE LIST 174

H. WAVE CLOCK PARAMETERS 180

APPENDIX F: PROGRAM LISTING 181

LIST OF REFERENCES 200

BIBLIOGRAPHY 201

INITIAL DISTRIBUTION LIST 202

LIST OF TABLES

I. Input Parameters for Simplified Test 39
II. Results of Simplified Data Set Run 40
III. Base Case for CAAM and SHIPSHOR Comparisons . . . 41
IV. Input Data for the Base Case 58
V. Sensitivity Index: Helicopter Time Factors 60
VI. Sensitivity Index: Beach Size 62
VII. Sensitivity Index: Landing Zone Size 64
VIII. Sensitivity Index: Operational Deck Spots 66
IX. Sensitivity Index: # of 6 Helo Sized LZs 68
X. Sensitivity Index: # of 4 Helo Sized LZs 70
XI. Sensitivity Index: Helicopter Launch Distance . . 72
XII. Sensitivity Index: Number of Helicopters 74
XIII. Sensitivity Index: Number of CH-46 Helicopters . . 76
XIV. Sensitivity Index: Number of Beaches 78
XV. Sensitivity Index: Landing Craft Launch Distance 80
XVI. Altered Parameters Yielding Greatest Effect in MCE 82
XVII. Summary of Input Data Sources 119
XVIII. Helicopter Time factors in Minutes 122
XIX. Boat Time Factors in Minutes 123
XX. Operational Deck Spots 124
XXI. Interactive Parameters 125
XXII. Helicopter Time Factors in Minutes 160
XXIII. Boat Time Factors in Minutes 161
XXIV. Operational Deck Spots 162
XXV. Interactive Parameters 164

LIST OF FIGURES

1.1 Purpose of Simulation 13
1.2 Goal of an Operational Model 13
2.1 General Design Process for a Simulation Model 18
3.1 Amphibious Task Force Areas 25
3.2 Boat Landing Process 27
3.3 Helo Flight Lanes 29
3.4 Helo Flight Plan 29
4.1 Flow Chart of the Model 31
5.1 Number of Helicopters: CAAM 42
5.2 Number of Helicopters: SHIPSHOR 43
5.3 Helicopter Launch Distance: CAAM 44
5.4 Helicopter Launch Distance: SHIPSHOR 45
5.5 Operational Deck Spots: CAAM 46
5.6 Operational Deck Spots: SHIPSHOR 47
5.7 Mix of Launch Distances and # Helos: CAAM 48
5.8 Mix of Launch Distances and # Helos: SHIPSHOR 49
5.9 Base Case 50
5.10 1.5 Wing and 1.0 Wing 51
5.11 15 NM Launch Distance and 35 NM Launch Distance 52
5.12 57 and 176 Operational Deck Spots 53
5.13 15 NM/1.0 Wing and 35 NM/1.0 Wing Distance/Number Mix 54
5.14 Helicopter Time Factors 59
5.15 Beach Size 61
5.16 Landing Zone Size 63

5.17 Operational Deck Spots 65
5.18 Number of 6 Helicopter Sized Landing Zones 67
5.19 Number of 4 Helicopter Sized Landing Zones 69
5.20 Helicopter Launch Distance 71
5.21 Number of Helicopters 73
5.22 Number of CH-46 Helicopters 75
5.23 Number of Beaches 77
5.24 Landing Craft Launch Distance 79
B.1 Flow Chart of Model Routines 94
B.2 Example of Landing Plan 95
B.3 Example of Data File T-1.DAT 96
B.4 Example of Interactive/Line Input Data 96
B.5 Example of a Graph of Build-up Ashore 97
C.1 Flow Chart of the System 104
C.2 Flow Chart of the Model 107
C.3 Graph of Build-up Ashore 108
C.4 Example of Landing Plan Data File 118
C.5 Example of Data File T-1.DAT 118
C.6 Example of Interactive/Line Input Data 119
C.7 Format for the Landing Plan File 121
C.8 Example of Output 127
C.9 Example of Output File 128
D.1 Flow Chart of the Model 132
D.2 Flow Diagram of HRLD/BRLD 140
D.3 Flow Diagram of HLNCH/BLNCH 143
D.4 Flow Diagram of HUNLD/BUNLD 146
D.5 Flow Diagram of HRTN/BRTN 148
D.6 Flow Diagram of HARIV/BARIV 150
D.7 Example of Landing Plan Data File 158
D.8 Example of Data File T-1.DAT 159
D.9 Example of Interactive/Line Input Data 163
E.1 Help File Listing 166

I. BACKGROUND

A. THE PROBLEM

"How would increasing the number of amphibious ships impact on an amphibious landing?", "what happens if ...?","suppose we only had x number of troop helicopters...?". One way to answer questions like these is to actually conduct landings under the different conditions desired. However, the cost of a single landing exercise is enormous, and to run many operations would quickly devour the entire defense budget. The next best option for obtaining answers to the problem above is the use of one of the many tools available in a large class of decision support aids. That instrument, which has become extremely important when used properly, is the computer simulation.

Before continuing, it is necessary to clarify some terms, which have over time, acquired different connotations. A model is a representation of an entity or situation by something else that has relevant features or properties of the original [Ref. 1: p.10]. Simulation is an analytic technique using the mathematical and logical models which represent a real system in order to study the processes over periods of time. Neither definition implies the use of a computer, however, for the purposes of this text the terms will carry that implication [Ref. 1: p.9]. Modelling is the process by which an analyst arrives at a model or simulation [Ref. 2: p.9].

A model or simulation can be distinguished by its ultimate purpose. Listed in Figure 1.1 [Ref. 1: p.10] are some of the more common ones.

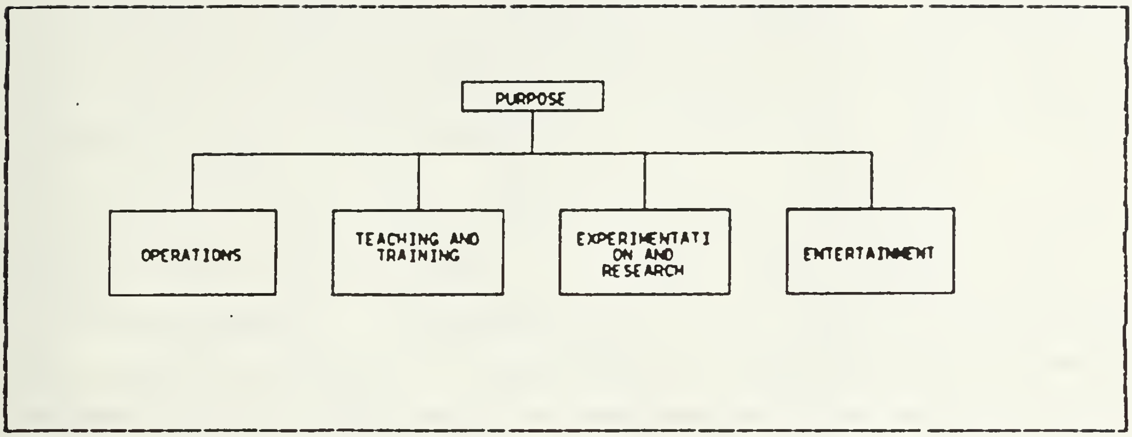


Figure 1.1 Purpose of Simulation.

The military applications of simulations are primarily concentrated in the operations category. The goals of an operational type model are broken down as in Figure 1.2 [Ref. 1: p.13].

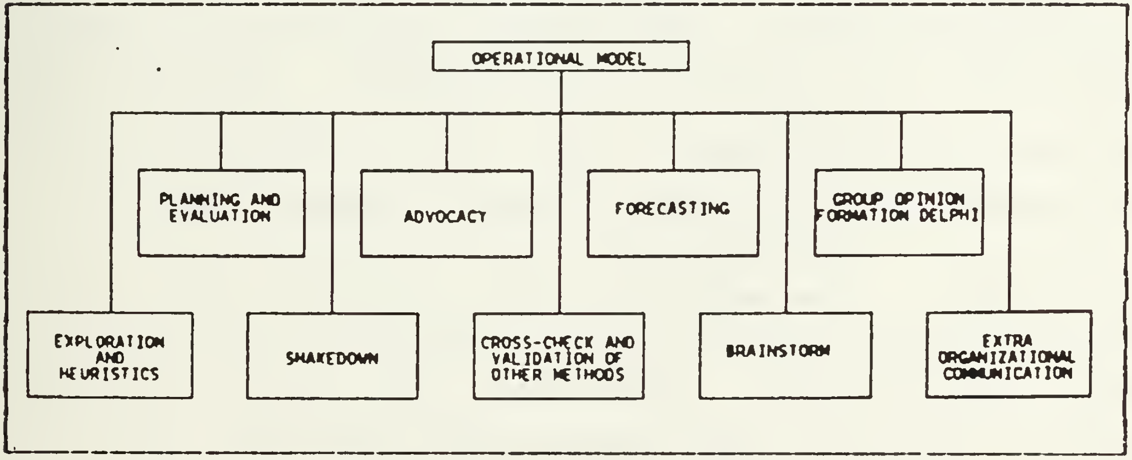

OPERATIONAL MODEL
PLANNING AND EVALUATION
ADVOCACY
FORECASTING
GROUP OPINION FORMATION DELPHI
EXPLORATION AND HEURISTICS
SHAKEDOWN
CROSS-CHECK AND VALIDATION OF OTHER METHODS
BRAINSTORM
EXTRA ORGANIZATIONAL COMMUNICATION

Figure 1.2 Goal of an Operational Model.

Of the goals listed above, the most important one for military purposes is the planning and evaluation category. Some of the most prevalent uses under planning and evaluation are technical evaluation, doctrinal evaluation, force structure evaluation, and planning. These classifications of simulations determine the structure of the model. A model designed for research, for instance, will be geared toward a technically oriented person who is familiar with computers, programming and the associated logic. A military oriented planning model must be designed for the staff officer who will be, more likely than not, unfamiliar with computers and programming techniques. This would imply the model should be easy to use and be accompanied by the appropriate level documentaion for the non-technical staff officer or decision maker. [Ref. 1: p.13]

The scenario illustrated in the first paragraph of this thesis was surely posed to some staff officer at Headquarters Marine Corps. The response was probably either to look for some existing model or contract to have one developed. The result is that there are several programs in existence which simulate the ship-to-shore movement of the amphibious operation. They exist as a portion of a larger combat model or as an entity by themselves. These models, having been written some time ago, were developed for use on large main frame computers. That fact automatically limits the availability to the staff officer or decision maker.

With the proliferation of microcomputers, the next logical step in the evolution of this type of model was to provide the staff officer with the ability to run the simulation from his desk. This would allow him to respond rapidly to 'what if' type questions which are often bantered about in planning sections of any headquarters. That requirement was, in fact, made and a civilian contract was let to develop a ship-to-shore model to be run on a microcomputer. Unfortunately, for some reason unknown to the author, the program that was developed was not implemented. It was 'shelved' at the Development Center, Quantico, Va. in an unusable form with little documentation. The author, during his experience tour was exposed to the model and was made aware of the need to have it completed. That leads in to the purpose of this thesis.

B. PURPOSE

The purpose of this thesis is to complete the development of the amphibious ship-to-shore model for use on a microcomputer. This will be accomplished by using a design process for simulations as if the model was being initiated from scratch. Included will be the capabilities and limitations of the model to ensure the decision maker has full knowledge of the output. Finally, the finished product will be accompanied by the appropriate level of documentation directed at the primary user.

II. SIMULATIONS

The process by which an analyst derives a model is considered as much an art as a science. Thus, there are no theorems or strict rules to guide one in the development. The actual steps followed by the model builder are very situation dependent. However, there are some basic techniques which are commonly followed by experienced modellers and provide a framework for good model development.

The approach begins with a simple model and attempts to evolve into a more elaborate model which more closely resembles the actual situation being modelled. As the model achieves the simplified goals, new problems may be identified or greater realism may be desired which would lead to revisions and additions. Thus, the evoution of a model never really stops until it has outlived its usefulness.

Based on the definition of simulation in chapter one, a simulation: (1) is concerned with the operation of systems; (2) is concerned with the solution of real world problems; and (3) is performed as a service for those interested in its behavior. From these characteristics the criteria for a good simulation are:

1. Simple to understand by the user
2. Goal or purpose directed
3. Robust, i.e., it does not give absurd answers when minor changes in parameters are made
4. Easy for the user to control and manipulate
5. Complete on important issues
6. Adaptive, with an easy procedure for modification or update
7. Evolutionary - start simple and become more complex in conjunction with the user's requirements [Ref. 3: p.22]

With the above criteria in mind, Figure 2.1 [Ref. 3: p.24] illustrates a general design process for a simulation model. Note the flows which return to earlier stages on the diagram. Those indicate the iterative process of the design. The following discussion will elaborate on each step. However, keep in mind the general nature of the process.

The most critical step in the development of a model of a system is to define the problem. Albert Einstein once stated that proper formulation of a problem was more important than its solution [Ref. 3: p.25]. One of the most important functions of an analyst is to translate the problem, often vaguely stated by the decision maker, into precise and operational terms [Ref. 4: p.51]. The analysis begins with specification of the goals and objectives and the establishment of boundaries of the model. A flow diagram which neither oversimplifies nor carries too much detail is the next step in the problem formulation. One should construct the model around the questions to be answered rather than imitate the real system exactly. [Ref. 3: p.27]

At this point a decision must made whether simulation is the optimal technique for solving the stated problem. If a less complicated, more direct technique exists, it should be pursued in the name of efficiency and cost.

Once the decision to use simulation is made, the data requirements must be analyzed. This includes data regarding the input and output of the system as well as information about the various components of the system [Ref. 3: p.27]. Availibilty of input data for the system must be determined in addition to historical output data for use in validation of the model. In conjunction with the data analysis is the identification of major variables to be used in the system which immediately preceed the formulation of model subsystems.

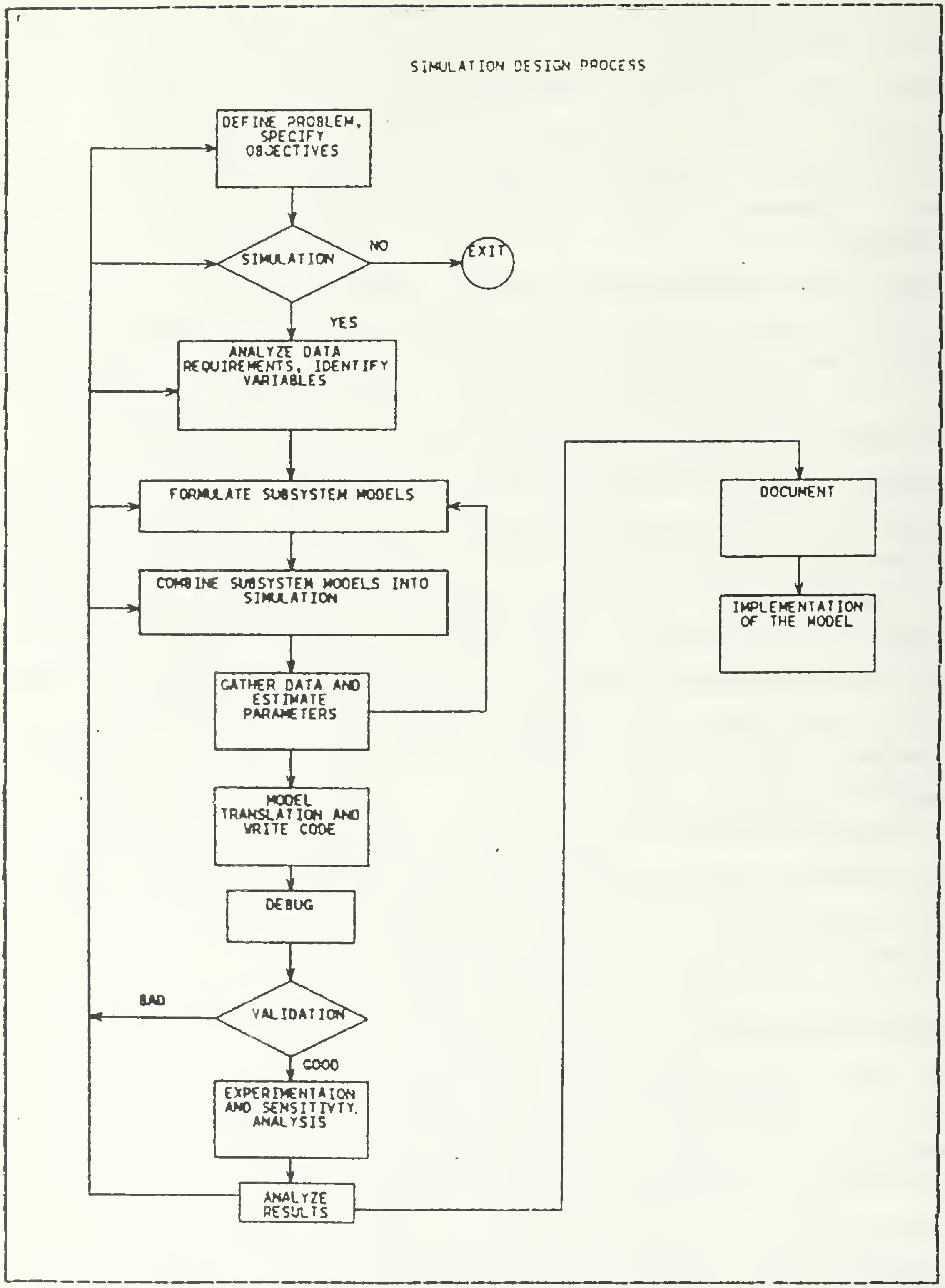

SIMULATION DESIGN PROCESS
DEFINE PROBLEM, SPECIFY OBJECTIVES
SIMULATION
NO
EXIT
YES
ANALYZE DATA REQUIREMENTS, IDENTIFY VARIABLES
FORMULATE SUBSYSTEM MODELS
COMBINE SUBSYSTEM MODELS INTO SIMULATION
GATHER DATA AND ESTIMATE PARAMETERS
MODEL TRANSLATION AND WRITE CODE
DEBUG
BAD
VALIDATION
GOOD
EXPERIMENTAION AND SENSITIVTY ANALYSIS
ANALYZE RESULTS
DOCUMENT
IMPLEMENTATION OF THE MODEL

Figure 2.1   General Design Process for a Simulation Model.

Large problems should be segmented into subsystems which are as independent of each other as possible. This facilitates a logical development and makes validation much less complicated. Without the submodels the entire simulation model would have to be formulated and validated at once - a formidible task. The decision maker should take part in this phase of development to further his knowledge of the model and the results and to keep the builder from straying from the purpose of the model. [Ref. 4: p.55]

After combining the subsystems into the simulation, the data and parameters identified ealier should be gathered to ensure they fit into the model. If necessary some subsystems may have to be adjusted to accomodate the data and parameters. The next stage involves choosing an appropriate computer language with the following considerations:

1. Efficiency of computer translation and running speed
2. The difficulty of translating the description of the subsystems into the computer language
3. The output desired from the simulation and ease of obtaining the output
4. The limitations of the computer hardware to use the language

Model translation, coding and debugging the model follow. The advantage to using submodels becomes very evident during the debug phase. The more independent the submodels are the easier it will be to debug on a component basis.

An important step in the design process is the validation of the simulation model. Validation is the process of bringing to an acceptable level the user's cofidence on how well the real system is being modelled. It is impossible to prove that any model is a "true" depiction of a real system, although a careful and thorough evaluation can build confidence in the simulation. [Ref. 3: p.29] This area is very deficient in simulation models found in DOD. Empirical tests are questionable or non-existent, and few efforts are being made to require proper validation. [Ref. 1: p.189]

There are three tests that may be used to validate a model. First, results that are produced must appear to be reasonable based on experience. Second is to test any assumptions made about the system and third is to test the input-output transformations. [Ref. 3: p.29] A common method of validating the output is to compare it to historical data collected under similar conditions. If the two are close, the model can be accepted as a representation of the real system. The ultimate validation, however, is how well the model predicts the future or how well it performs the analysis it was designed to perform. [Ref. 4: p.57]

The amount and type of experimentation and sensitivity analysis depend on the main purpose of the simulation. At this stage, the model is run using various sets of data and parameters in a systematic manner to observe the response. Often it is at this point when flaws and oversights in planning become more evident requiring the builder to retrace some steps in the design process. Deficiences in estimating the validity of input parameters is a result of insufficient or poorly conducted sensitivity analysis. [Ref. 1: p.189]

The final two elements in the design process must be included in any simulation project. Implementation refers to refining the model, training the user, adjusting the model to changing conditions and ensuring that the results found are valid. [Ref. 3: p.33] Good documentation is extremely valuable for facilitating changes and gaining a basic understanding of the model. Resources are wasted as a result of poor documentation. This was noted by GAO reports which investigated many of the problems found with operating or modifying models. Even though implementaion and documentaion are tedious and difficult, when done properly they can extend the life of a model. More importantly, they facilitate the use of simulation which is particularly crucial in an environment heavily oriented toward the need of users. [Ref. 1: p.191]

In the chapters which follow, the procedure outlined above will be modified slightly to fit the conditons under which the actual model development took place. Each phase of the design process will be made apparent during the presentation as well as any deviations from those steps as explained above.

III. THE SYSTEM

An amphibious operation is an attack launched from the sea by naval and landing forces embarked in ships or craft involving a landing on a hostile shore. The principle type of amphibious operation is the amphibious assault which is distiguished from other types in that it involves establishing a force on a hostile shore. The amphibious assault follows a sequence of activities which consist of planning, embarkation, rehearsal, movement to the objective, and finally, assault and capture of the objective. [Ref. 5: p.1-3]

The ship-to-shore movement is the part of the assault phase which pertains to the timely deployment of troops and equipment from assault shipping to designated positions ashore on the landing area. This movement is designed to ensure the landing of troops, equipment , and supplies at the prescribed times and places in such a manner as to support the scheme of maneuver for operations ashore. The movement can be accomplished by waterborne means, by helicopters, or by a combination of the two. [Ref. 6: p.45]

The ship-to-shore movement is the most critical part of the assault phase. The coordination and control of the many diversified naval and troop elements requires a high degree of very detailed planning. Such items as the landing craft and helicopter availability, amphibious vehicle and helicopter employment, assault schedule, landing sequence, serial assignments (a serial is a number assigned to an element of the landing force), and wave assignments are just a few of the areas to be planned prior to execution of the landing. [Ref. 5: p.11-7]

The ship-to-shore movement is further divided into two periods, initial unloading and general unloading. The initial unloading period is primarily tactical in nature and therefore must be responsive to the landing force requirements ashore. A tactical situation is where the troops and equipment going ashore are prepared to fight. They are organized in combat units and execute a set plan for the purpose of forcibly taking the beach and landing zones. It is during this period that the landing force is establishing a foothold ashore and the unloading of units is very selective. The general unloading period is logistical in nature and involves the timely delivery of needed supplies and equipment. During this time period, the majority of combat troops are established ashore and are involved with the execution of the operational plan of attack. A beachhead exists and is considered a more secure rear area where supplies and equipment continue to arrive in support of the combat troops at the front.

There are several descriptive categories of waves[1] involved in the movement of forces and equipment ashore according to the landing plan. These categories actually divide the entire landing force and their equipment into major groups to provide flexibility, organization, and control of the landing. Three of these categories are concerned with troops and their initial combat supplies. These are scheduled, on-call, and nonscheduled waves.

Two other categories of waves are are concerned with unloading of supplies. These are called floating dumps and landing force supplies which only apply to the water-borne means of delivery. They are not involved with the actual initial combat landing.

---

[1] A wave consists of a number of landing craft or helicopters which are to be landed simultaneously. The size of a wave is limited by beach or landing zone capacity and the availibility of vehicles. Tactical unit integrity is paramount when constructing a wave.

The water-borne movement is broken down into all five categories. These include:

1. Scheduled waves --Scheduled waves are formations which consist of landing ships, landing craft, or amphibious vehicles which contain troops, equipment, and supplies to be landed at a predetermined time and place. Their landing begins at H-hour and continues for a relatively short period of time.
2. On-call waves --On-call waves include landing ships, amphibious vehicles, or landing craft containing troops, equipment, and supplies for which an early need ashore is anticipated, but for which the time or place cannot be accuately predicted. The commander will call for these waves at the desired time and place depending on the situation ashore.
3. Nonscheduled units --Nonscheduled units are those landing force elements and supplies held in readiness during the initial unloading, but not included in scheduled or on-call waves. The probable sequence of landing nonscheduled units is determined during planning, however, the landing of these units is directed from shore when the situation dictates.
4. Floating dumps --Floating dumps are emergency supplies preloaded in landing craft, amphibious vehicles, or in landing ships.
5. Landing force supplies --Landing force supplies are those supplies remaining on assault shipping after initial combat supplies and floating dumps have been unloaded.

As the amphibious force moves into the objective area the different types of ships move into various sea areas. Sea areas are used to minimize possible interference between various components of the amphibious task force and other supporting forces. Figure 3.1 [Ref. 5] shows these areas.

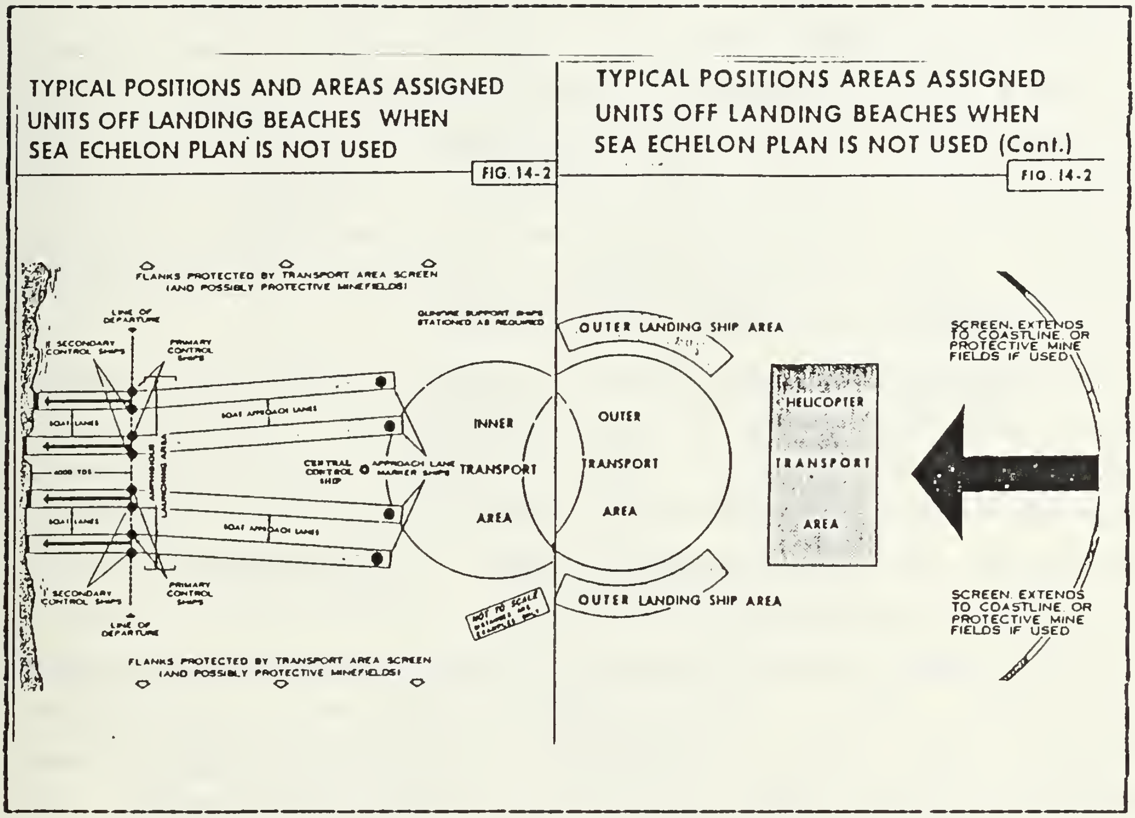

TYPICAL POSITIONS AND AREAS ASSIGNED UNITS OFF LANDING BEACHES WHEN SEA ECHELON PLAN IS NOT USED
FIG. 14-2
TYPICAL POSITIONS AREAS ASSIGNED UNITS OFF LANDING BEACHES WHEN SEA ECHELON PLAN IS NOT USED (Cont.)
FIG. 14-2
FLANKS PROTECTED BY TRANSPORT AREA SCREEN (AND POSSIBLY PROTECTIVE MINEFIELDS)
LINE OF DEPARTURE
GUNFIRE SUPPORT SHIPS STATIONED AS REQUIRED
SECONDARY CONTROL SHIPS
PRIMARY CONTROL SHIPS
BOAT LANES
BOAT APPROACH LANES
4000 YDS
AMPHIBIOUS LAUNCHING AREA
CENTRAL CONTROL SHIP
APPROACH LANE MARKER SHIPS
INNER TRANSPORT AREA
NOT TO SCALE
OUTER LANDING SHIP AREA
OUTER TRANSPORT AREA
HELICOPTER TRANSPORT AREA
SCREEN EXTENDS TO COASTLINE OR PROTECTIVE MINE FIELDS IF USED

Figure 3.1 Amphibious Task Force Areas.

Some definitions of the various sea areas follow. The LOD (line of departure) is a designated line off-shore approximately parallel to the beach. From this line, successive boat waves are dispatched for their final movement to the beach. The waves travel in boat lanes which extend seaward from the beach to the LOD and are the same width as the beach. Approach lanes are extensions of the boat lanes from the LOD to the transport area.

Landing ships are initially staged in the outer landing area and when instructed, move to the amphibious launching area. This is located seaward of the LOD and parallel to the beach. Landing ships move in this area parallel to the LOD at about 15 knots and discharge the amphibious vehicles which make up the initial assault waves. (see Figure 3.1)

Transports move from the outer transport area, where they were initially staged, into the inner transport area. This area is as close to the beach as the hydrography[2] and enemy situation will permit since boat operations must be conducted at a very slow speed or at a dead stop. Landing craft are loaded and launched from this area and proceed to the LCD via approach lanes. (see Figure 3.1)

All amphibious vehicles and landing craft move to the LOD at times designated by the landing plan. Movement from the LCD is controlled by various control officers located along the way and waves move according to their category (scheduled, on-call, nonscheduled) at appropriate times. Those times have been computed based on the number of vehicles the beach can accomodate, speed of the vehicles, scheme of maneuver, etc. If the wave is comprised of amphibious vehicles there is no turnaround, which means that those vehicles proceed inland and do not return to the ship. If the wave is made up of boats, after unloading they return to the transport areas for subsequent loads. Figure 3.2 is a diagram of the process.

The helicopter movement is broken down into three groups. This movement is normally completed during the initial unloading period when tactical[3] considerations are the dominant factors. Once the helicopter-borne movement is completed, transport helicopters are employed for tactical

---

[2] Hydrography is the description of the physical conditions of a body of such as the depth, type of bottom (sandy, rocky), currents, etc.

[3] Tactical refers to a high state of combat readiness maintained while executing a combat plan. The assumption is made that there will be contact with enemy forces.

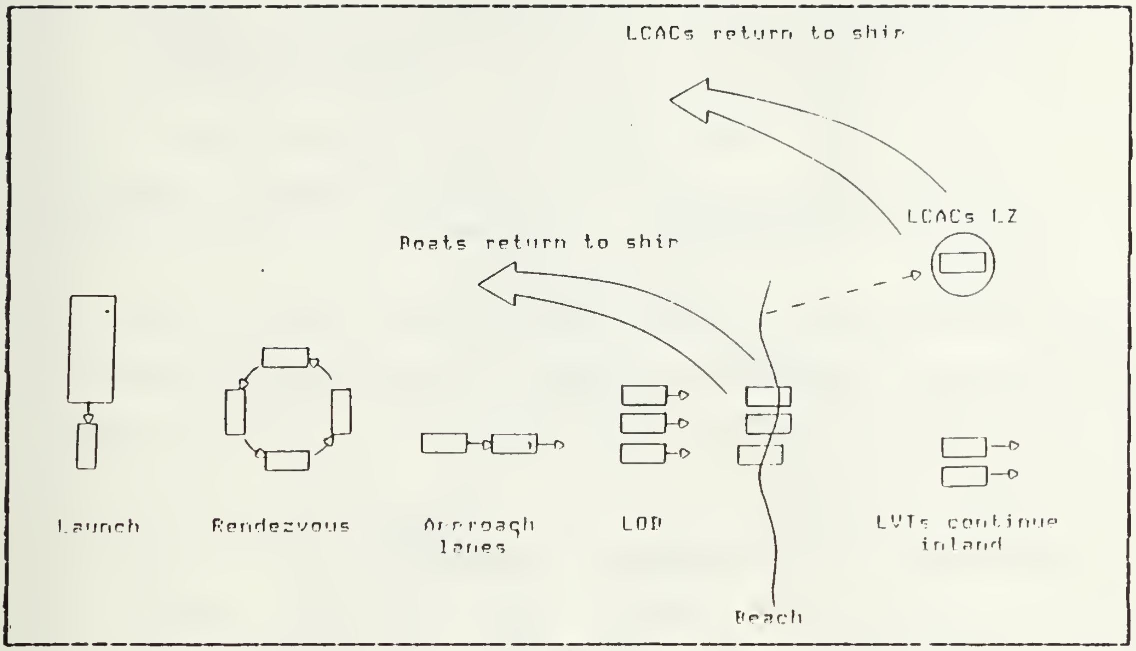


Figure 3.2 Boat Landing Process.

and logistical requirements as needed by the landing force. Categories for the air movement include:

1. *Scheduled waves* --This category consists of those assault elements of the landing force, together with their initial combat supplies, which will be landed by helicopter. Times, places, and formations for landing have been determined. Scheduled landing begins at a specified time and continues until all elements in this category are landed.
2. *On-call waves* --This category consists of those helicopterborne units with their initial combat supplies, or emergency supplies, whose need ashore at an early hour is anticipated, but whose time and place of landing cannot be accurately predicted. On-call waves may interrupt other landing categories based on the urgency of the requirement. On-call elements are held in readiness aboard shipping and are landed when called by the appropriate level commander.

3. Nonscheduled units --Nonscheduled units consist of any remaining units of the landing force and their initial combat supplies which have not been included in either the scheduled or on-call categories. This category can be interrupted or temporarily suspended because of unforeseen circumstances in the situation ashore.

The shipping carrying the helicopters initially moves to the helicopter transport area which is generally seaward of the outer transport area. (see Figure 3.1) From this area, helicopters comprising the first assault waves are readied and spotted on the flight decks. On signal, troops enplane, and helicopters are launched. Flights of helicopters rendezvous about their parent ship and then proceed as predetermined waves. They travel via designated helicopter lanes and control points enroute to the landing zone. Figure 3.3 [Ref. 7: p.143] is a diagram of this process. The helicopter waves pick up attack helicopter protection that provides ground fire suppression. The waves land, are unloaded, and return to the ships to refuel and enplane subsequent troop serials.[4] A schematic of this process is found in Figure 3.4 [Ref. 7: p.121].

LCAC (landing craft air cushion) would launch from greater distances than boats because of their speed. They would operate much like helicopters in that they can travel inland to off load troops and equipment instead of stopping at the beach. They would also be able to return to shipping for subsequent loads.

---

[4] A serial is a designator (number) given to a combat unit or a piece of equipment for the purpose of assignment to a carrier going to shore. This provides control and organization during the landing phase of the operation. A wave is made up of many serials.

ENCLOSURE 2 (Helicopter Landing Diagram) to Tab D (Vertical Assault Plan) to Appendix 5 (Landing Force Landing Plan) to Annex R (Amphibious Operations) to IV MAF OPLAN 9999 (U)

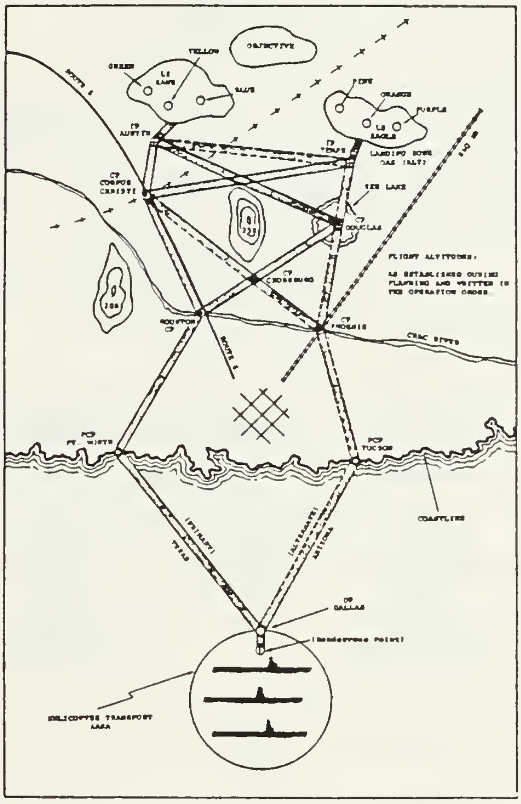


Figure 3.3 Helo Flight Lanes.

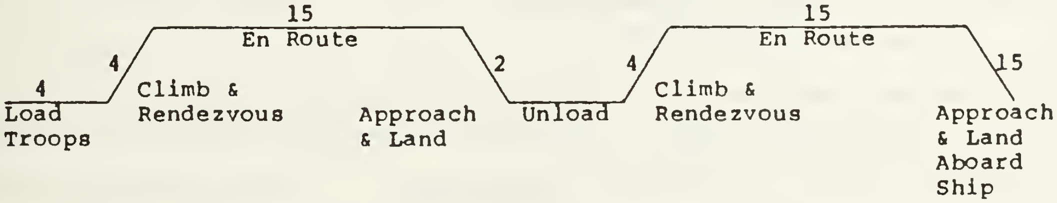


Figure 3.4 Helo Flight Plan.

IV. THE MODEL

The ship-to-shore model attempts to simulate the process described in chapter 3 of this paper. Preparation of the model for use involves construction of the serials and assault waves which is done manually as they would be prior to an operation. This process is the basis for the largest of three data files used in the simulation. Unfortunately, construction of the serials and waves is a very time consuming process and has not been automated at this time.

Two other data sources are needed to run the model. One file contains parameter values which are not subject to change. A helicopter time factors table which provides expected times for various evolutions such as loading or unloading is an example of the type of data found in this file. The last source of data is found within the program itself as either a line of code or as an interactive query. This data is of the type that may be changed easily for various types of analysis.

The model utilizes 35 subroutines to attempt to translate reality into mathematical computations. A diagram of the model in very general terms is found in Figure 4.1 The first subroutine in the program reads the data from the wave and serial file and the parameter file. Also the variable data is entered within the first section of the program. The second subroutine initializes the arrays and clocks to be used by the timing and activity subroutines based on the information found the wave data file. These subroutines essentially complete the set-up and initialization of the model.

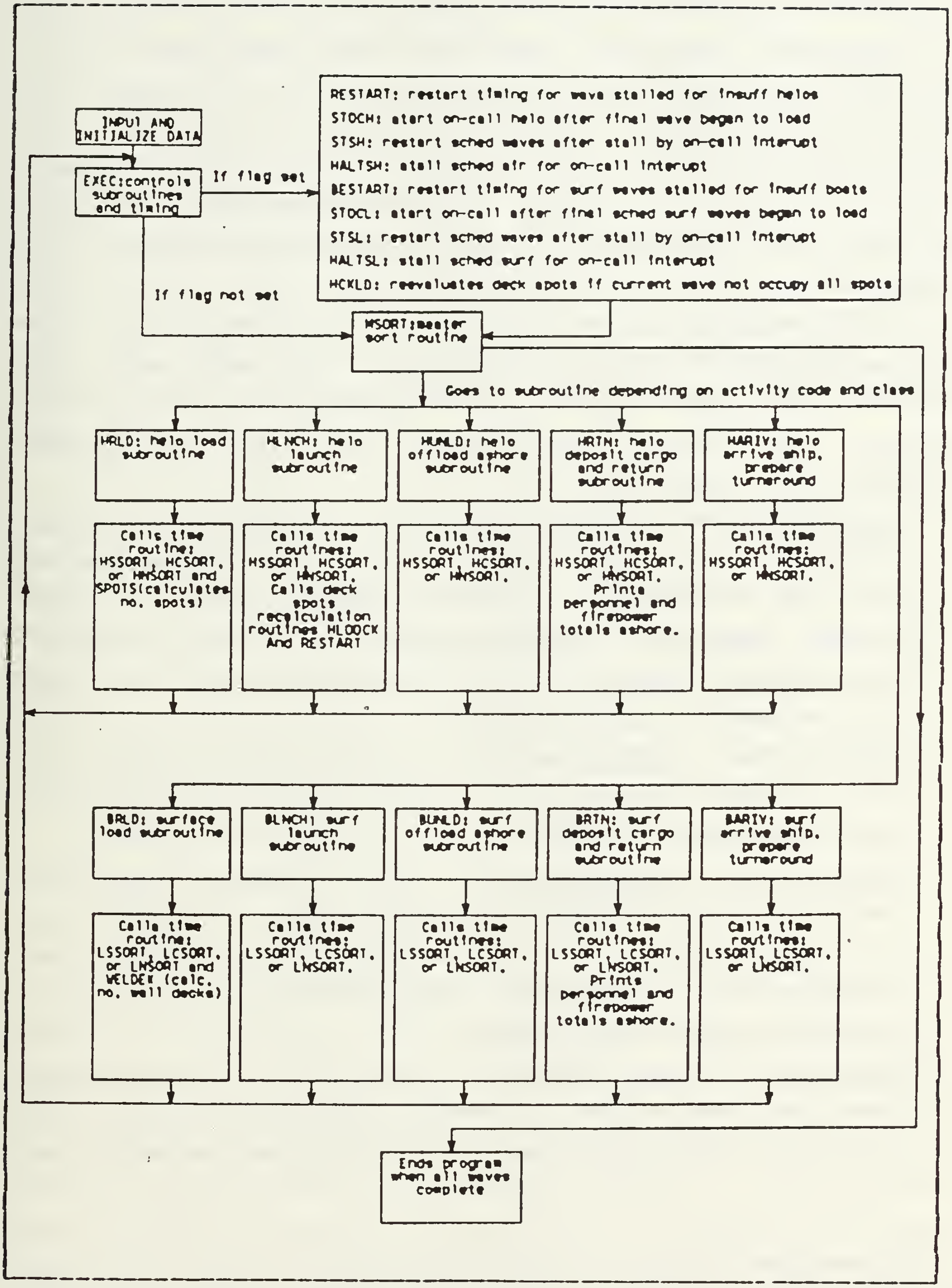

INPUT AND INITIALIZE DATA
EXEC:controls subroutines and timing
If flag set
RESTART: restart timing for wave stalled for insuff helos
STOCH: start on-call helo after final wave began to load
STSH: restart sched waves after stall by on-call interupt
HALTSH: stall sched air for on-call interupt
BESTART: restart timing for surf waves stalled for insuff boats
STOCL: start on-call after final sched surf waves began to load
STSL: restart sched waves after stall by on-call interupt
HALTSL: stall sched surf for on-call interupt
HCKLD: reevaluates deck spots if current wave not occupy all spots
If flag not set
MSORT:master sort routine
Goes to subroutine depending on activity code and class
HRLD: helo load subroutine
HLNCH: helo launch subroutine
HUNLD: helo offload ashore subroutine
HRTN: helo deposit cargo and return subroutine
HARIV: helo arrive ship, prepare turnaround
Calls time routine: HSSORT, HCSORT, or HNSORT and SPOTS(calculates no. spots)
Calls time routines: HSSORT, HCSORT, or HNSORT. Calls deck spots recalculation routines HLOOCK And RESTART
Calls time routines: HSSORT, HCSORT, or HNSORT.
Calls time routines: HSSORT, HCSORT, or HNSORT. Prints personnel and firepower totals ashore.
Calls time routines: HSSORT, HCSORT, or HNSORT.
BRLD: surface load subroutine
BLNCH: surf launch subroutine
BUNLD: surf offload ashore subroutine
BRTN: surf deposit cargo and return subroutine
BARIV: surf arrive ship, prepare turnaround
Calls time routine: LSSORT, LCSORT, or LNSORT and WELDEK (calc. no. well decks)
Calls time routines: LSSORT, LCSORT, or LNSORT.
Calls time routines: LSSORT, LCSORT, or LNSORT.
Calls time routines: LSSORT, LCSORT, or LNSORT. Prints personnel and firepower totals ashore.
Calls time routines: LSSORT, LCSORT, or LNSORT.
Ends program when all waves complete

Figure 4.1 Flow Chart of the Model.

Once the model enters the EXEC subroutine it continues the time/event[5] procedure until all waves have been landed. The EXEC subroutine is the control routine in which all waves eventually pass. Based on the class of the wave (air or surface) and the current activity (load, launch, unload, return, or arrive back ship), the EXEC subroutine will send the wave in question to the proper subroutine for manipulation.

Prior to going to one of the activity subroutines, the program queries whether certain flags have been set. If they have been set, one of the following routines could be called. Five are for helicopter waves and four are for surface waves. They are:

- RESTART(helo) / BESTART(surface): Restart timing for waves that have been stalled by the lack of deck spots (helo or surface). Since in reality there are not enough deck spots to accomodate all of the assault helicopters (landing craft), subsequent waves must wait for sufficient spots to load.
- STOCH(helo) / STOCL(surface): These subroutines start the on-call waves to load after the final scheduled wave has begun to load.
- HALTSH(helo) / HALTSL(surf): Stall remaining scheduled waves when one or more on-call waves interrupt the scheduled category. As in reality, on-call waves can break in when called for. The model allows flags to be set to simulate this process.
- STSH(helo) / STSL(surface): Start scheduled waves that have been stalled due to on-call waves interrupting the sequence. More than one on-call wave can interrupt at

---

[5] Time/event refers to the computation of time required to complete a certain event such as loading a helicopter. That time is added to a continuous running clock to determine the time at which that event will take place.

different times so the HALTSH(L)/STSL(L) subroutines can be called more than once.

- HCKID(helo): Reevaluates deck spot employment if the current wave does not occupy all spots available. It calls WSPOT which calculates the number of spots to be used in the current wave.

One last subroutine will be called prior to the activity subroutines. This is the MSORT routine which acts as the master sort routine. It takes the lowest time from six other time/event sort routines and that becomes the next event which is processed by the appropriate activity subroutine.

There are three time sort routines common to the helicopter activities and three time sort routines common to the surface activities. They all sort through the waves identifying the wave with the earliest time for the next event. That wave is placed on the top of the queue from which MSORT then picks the lowest time out of the six time sort routines. Listed below are the sort subroutines:

- HSSORT: Scheduled helicopter wave sort routine.
- HCSORT: On-call helicopter wave sort routine.
- HNSORT: Nonscheduled helicopter wave sort routine.
- ISSORT: Scheduled surface wave sort routine.
- ICSORT: On-call surface wave sort routine.
- INSORT: Nonscheduled surface wave sort routine.

The activity subroutines process each wave based on its class, activity code, and category (scheduled, on-call, or nonscheduled). There are five major activity subroutines for each class (air or surface). Each activity subroutine calls other secondary routines and the appropriate time sort routine as determined by the wave category. Once the wave has completed an activity subroutine, it returns to the EXEC subroutine and the cycle begins again. Thus, each wave will eventually cycle through five subroutines.

The following is a summary of the activity subroutine and their supporting routines:

- HRLD: The purpose is to load (or attempt to load) a helicopter wave from available flight deck spots of the shipping. Loading will not be initiated if (a) no deck spots are currently available or if (b) insufficient helicopters of the proper type to carry the serials are present at the time. HRID calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT) and subroutine SPOTS. The SPOTS routine identifies whether all the deck spots are curently being employed for loading. A flag is set to indicate whether deck spots are available or not.
- HLNCH: This routine launches a helicopter wave from ships and determines the time it will arrive at the landing zone(s). It determines the number of deck spots made available to subsequent waves and restores those spots. Also it reevaluates the loading time for a wave being loaded which had employed all of the remaining deck spots. HLNCH calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT) and HLODCK and RESTART. HLODCK sets up signals that will initiate a reevaluation of the deck spot employment in the event that the current wave was not employing the last available deck spot. Those signals will trigger the use of HCKID which is called from EXEC. If the current wave was occupying the remaining deck spots, a flag is set to call RESTART from EXEC. RESTART allows subsequent waves to employ deck spots which will be vacated by the current wave.
- HUNLD: HUNLD determines whether or not landing zones are still occupied by previous waves and if so delays the landing of the current wave until the pertinent zones are free to use. Also it calculates the length

of time necessary for this wave to unload (constrained by the size of the landing zone). HUNLD calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT).

- HRTN: The purpose is to print a running total of the personnel ashore and the firepower ashore. Also HRTN calculates the time that the wave will return to the area of the shipping and hence be available for formation of subsequent waves. It calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT).
- HARIV: HARIV returns helicopters to the helicopter pools for use in forming subsequent waves after degrading the number of helicopters to reflect maintenance losses. It also initiates the loading process for subsequent waves. HARIV calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT).

The surface activity subroutines operate in basically the same manner (the first letter for surface activities is B instead of H). Since the following routines are so similar to the helicopter routines, only the differences will be pointed out.

- BRLD: The purpose is to load the landing craft in a similar fashion to the HRLD for helicopters. It calls the appropriate time sort routines (LSSORT, LCSORT, or LNSORT) and the subroutine WELDEK. WELDEK calculates the well decks to be used by landing craft of each wave and indicates whether there are sufficient openings or not.
- BLNCH: This routine launches the surface waves from the ships. It calls the appropriate time sort routine (LSSORT, LCSORT, or LNSORT).
- BUNLD: BUNLD determines if the beach is open for landing and off-loads the wave if it is possible. It calls the appropriate time sort routine (LSSORT, LCSORT, or LNSORT).

- BRTN: BRTN prints the running total of personnel ashore and the firepower ashore. Also, it returns the boats to the ships. It calls the appropriate time sort routine (LSSORT, LCSORT, or LNSORT).
- BARIV: This routine returns the boats to the boat pools for subsequent use. It calls the appropriate time sort subroutine (LSSORT, LCSORT, or LNSORT).

Once each wave has cycled through the five helicopter activities or the five surface activities, they are assigned a very large value in their time array. This prevents them from recycling. The program will end when the last nonscheduled surface wave has gone through all activities. At that point, all personnel in the wave data file will be ashore.

V. VALIDATION AND VERIFICATION

A. GENERAL

The purpose of validation[6] of a model is to bring to an acceptable level the user's confidence on how well the real system is being modelled [Ref. 3: p.29]. A common method and perhaps the best is to compare the results of the model with historical data. However, after some research it was discovered that there is a lack of historical data concerning actual ship-to-shore operations in the Marine Corps. Although there is data collected for reports which are required after training exercises, none of the pertinant information is retained after the reports have been compiled.

The next best method available to verify this model was to compare the results to another sophisticated simulation designed and refined to run on a large main frame computer. Several such models exist and the results utilized in this thesis are found in a study which employed one of those models [Ref. 9].

This section will be divided into two major subsections for the purposes of organization. The first will be concerned with verification and will cover an initial phase using a simplified test data set and the second phase will use more complicated input data. The second major subsection will involve sensitivity analysis of the model.

---

[6] Although there are many authors who distinguish between "verification" and "validation" of a model, there is no consistent use of the terms in the literature. Therefore, the terms will be used interchangeably in this thesis and have the same meaning [Ref. 8: p.631].

B. VALIDATION

1. Initial Phase: Simplified Input File

The initial phase of the verification analysis is designed to ensure the model is operating correctly for a simplified case which can be computed by hand. A simple landing plan will not test all the routines within the program. However, the major routines will be exercised to demonstrate that the basic logic is working.

The input data for the model for this phase is located in Table I . As one can observe the landing plan is reduced to essential information only. Data file T-1.DAT remains unchanged and the interactive input provides the parameters necessary to run this test. At this point many acronyms will appear in the text and tables. Appendix A contains a glossary of those acronyms that will be encountered.

Using the data in the Helo Time Factors Table (Table XVIII), hand calculations were made to determine the time that each wave would land and unload in the LZ from the time the wave was loaded on the ship. Also the number of troops being unloaded was computed at this time. Table II gives the results of the model and those of the hand calculations.

It is obvious by the results of this extremely simple run that the model is at least functioning in a basic mode. To attempt to verify a complex landing plan by hand calculations would be ludicrous. Therefore the method alluded to in the introduction of this chapter, comparison with historical data, would be the best way to verify a large landing plan. However, since there is a lack of historical data, another more sophisticated simulation was employed.

TABLE I

Input Parameters for Simplified Test

Landing Plan Test

Wave/Serial	# Pers	Carrier Type	Number Carriers	Dest
Sched helo #1/#1	10	CH-46	2	R-1
Sched helo #1/#2	10	CH-46	2	R-1
Sched helo #2/#1	10	CH-46	2	R-1
Sched helo #2/#1	10	CH-46	2	R-1
Sched surf #1/#1	10	LCM-8	2	G-1
On-call helo #1/#1	10	CH-46	2	R-1
On-call surf #1/#1	10	LCM-8	2	G-1
Nonsched helo #1/#1	10	CH-46	2	R-1
Nonsched surf #1/#1	10	LCM-8	2	G-1

Input parameters	Values
Operational deck spots	60
Helicopter Launch distance	10 NM
Shore to landing zone distance	2 NM
Boat launch distance	8 NM
Landing zone size	Unlimited
Beach size	Unlimited
Number of carriers	
Helos	12
Boats	4
Boat launch hour	0
Helo launch hour	0

TABLE II

Results of Simplified Data Set Run

	Time of unload on LZ/beach		# personnel ashore	
Wave	Model	Hand	Model	Hand
Sched helo #1	21	21	20	20
Sched helo #2	27	27	40	40
On-call helo #1	29	29	50	50
Nonsched helo #1	38	38	60	60
On-call surf #1	55	55	70	70
Nonsched surf #1	55	55	80	80

2. Phase Two: Comparison to Large Scale Simulation

A computerized amphibious assault model (CAAM) was used in the study in [Ref. 9]. CAAM utilized a computer language called GPSS (General Purpose Simulation System) which is designed specifically to run simulations. Since the study was mainly concerned with the air portion of the amphibious landing, all of the results deal with helicopoter-borne assaults only. This is not a problem since SHIPSHOR has the ability to run in an all helicopter mode, all surface mode, or a combination of the two. Thus, for the purposes of the validation, only the helicopter operation of SHIPSHOR will be utilized. Although this is not considered an ideal situation, the results from other simulation models were not available.

Several sets of results were contained in the Amphibious Lift Factors Study. Four of these will be used to make the comparisons. An attempt was made to duplicate the CAAM inputs in order to reach some viable conclusions about the comparisons. Duplication was not possible in all cases. However, the input parameters that were used by SHIPSHOR came very close to those used in CAAM.

A case base was used as a standard for comparison. It consisted of a set of input parameters which produced a profile of the progress of the landing. The base case was then compared to other situations to obsevre the effect on the landing profile. Table III contains the base case.

TABLE III

Base Case for CAAM and SHIPSHOR Comparisons

	Number of Helicopters			
Launch Dist	CH-46	CH-53D	CH-53E	Attrition
50 NM	156	80	32	0

Speed (type load)		Operatonal	Number of
Internal	External	Deck Spots	LZ's
120 kts	100 kts	114	3

Beach to LZ Distance	Size of Landing Zone
5 NM	Infinite

The following section will present four situations involving different input parameters for both CAAM and SHIPSHOR. The parameters will be listed followed by the graphic presentation of each case.

a. CAAM vs SHIPSHOR: General Comparison

Number of Helicopters

1. Base Case: 156 CH-46, 80 CH-53D, 32 CH-53E
2. 1.5 Wing: 108 CH-46, 64 CH-53D, 32 CH-53E
3. 1.0 Wing: 72 CH-46, 48 CH-53D, 16 CH-53E

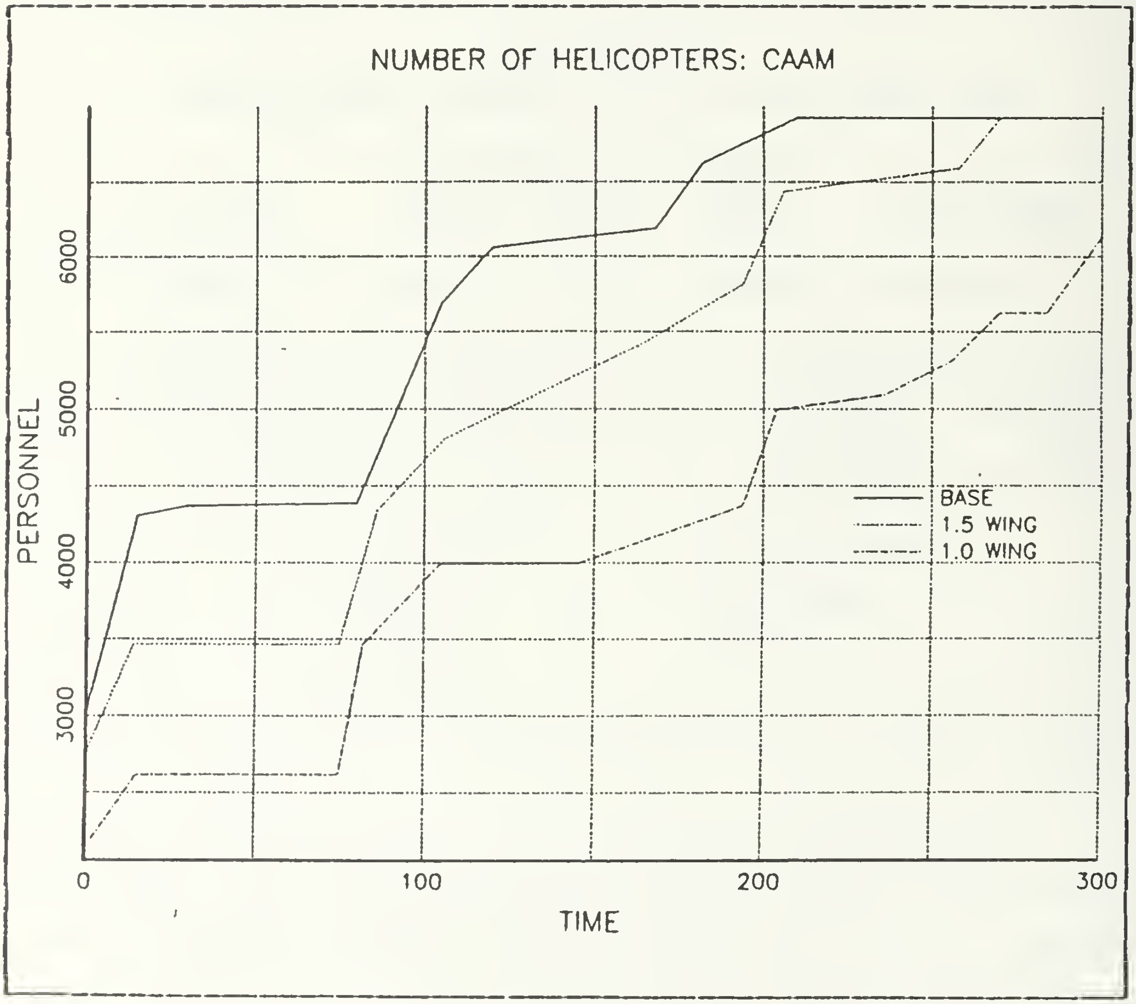


Figure 5.1 Number of Helicopters: CAAM.

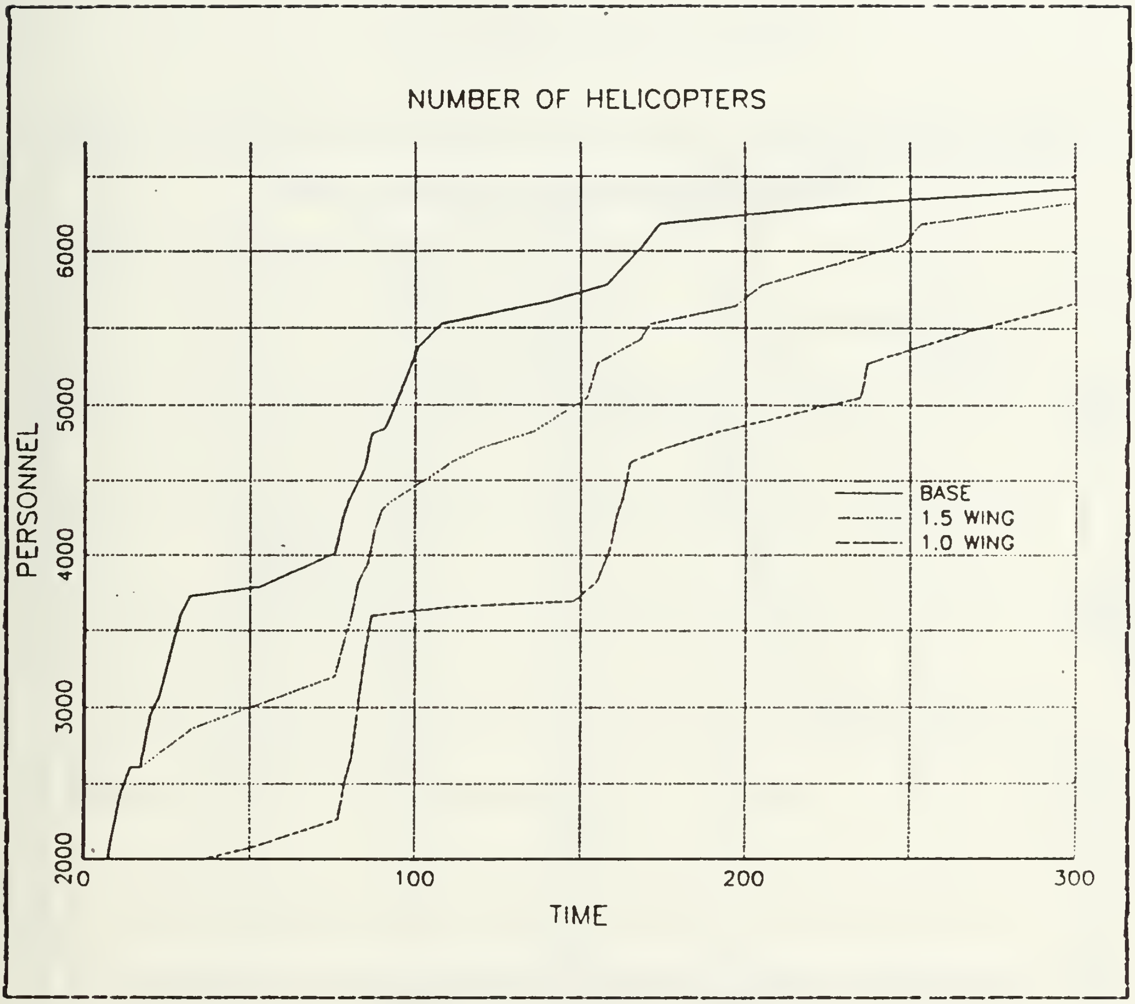


Figure 5.2 Number of Helicopters: SHIPSHOR.

Helicopter Launch Distance

1. Base case: 50 NM
2. Alternate case #1: 35 NM
3. Alternate case #2: 15 NM

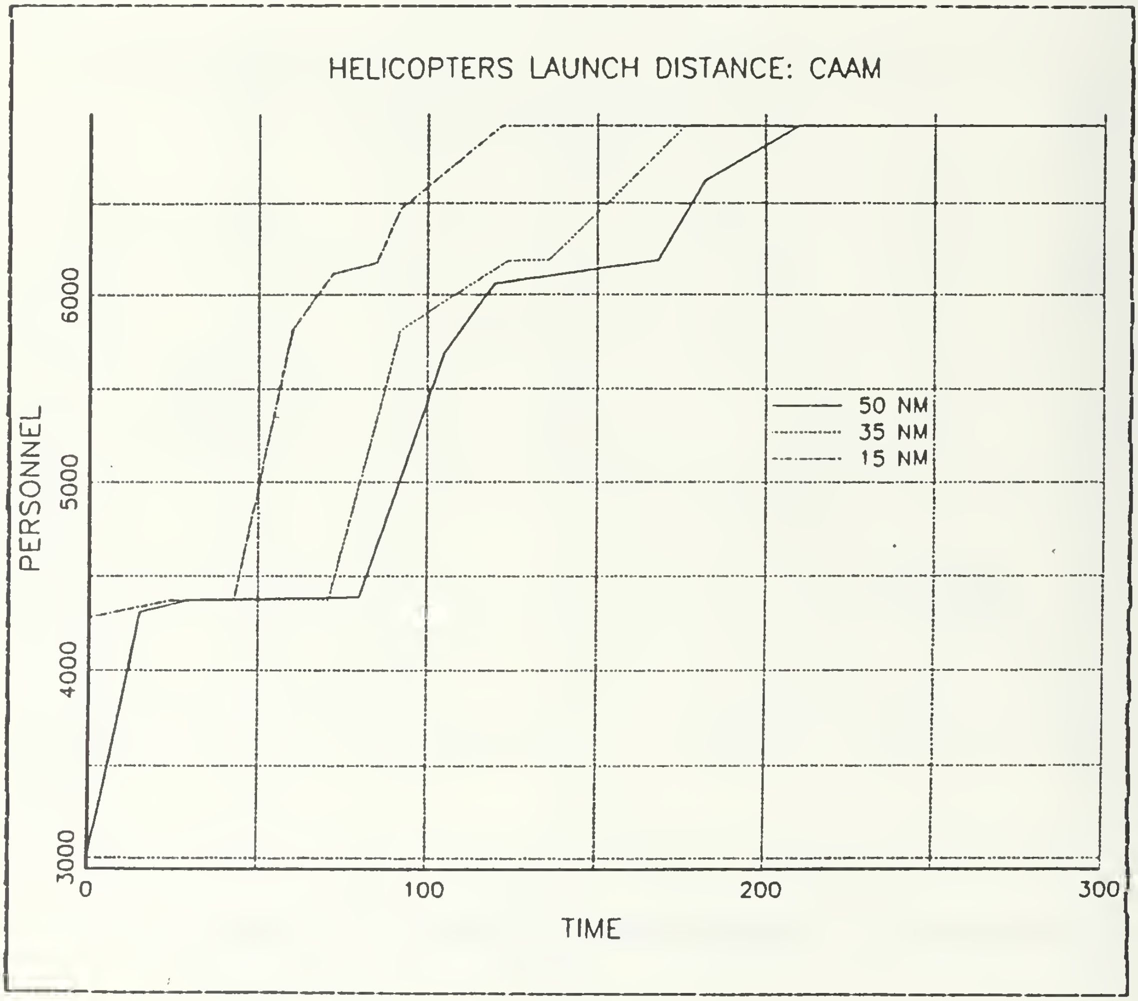


Figure 5.3 Helicopter Launch Distance: CAAM.

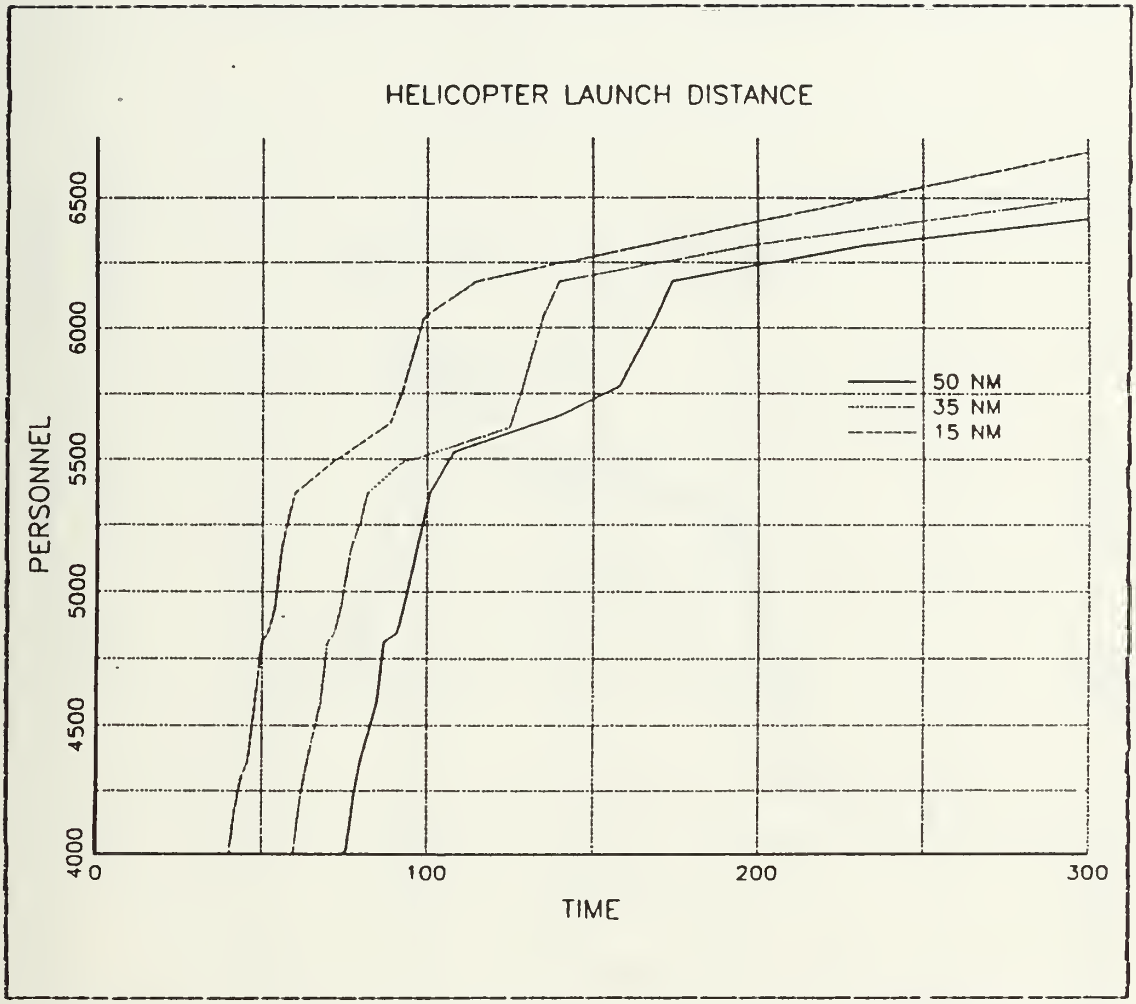


Figure 5.4 Helicopter Launch Distance: SHIPSHOR.

Operational Deck Spots

1. Base case: 114 operational deck spots
2. Alternate case #1: 176 operational deck spots
3. Alternate case #2: 57 operational deck spots

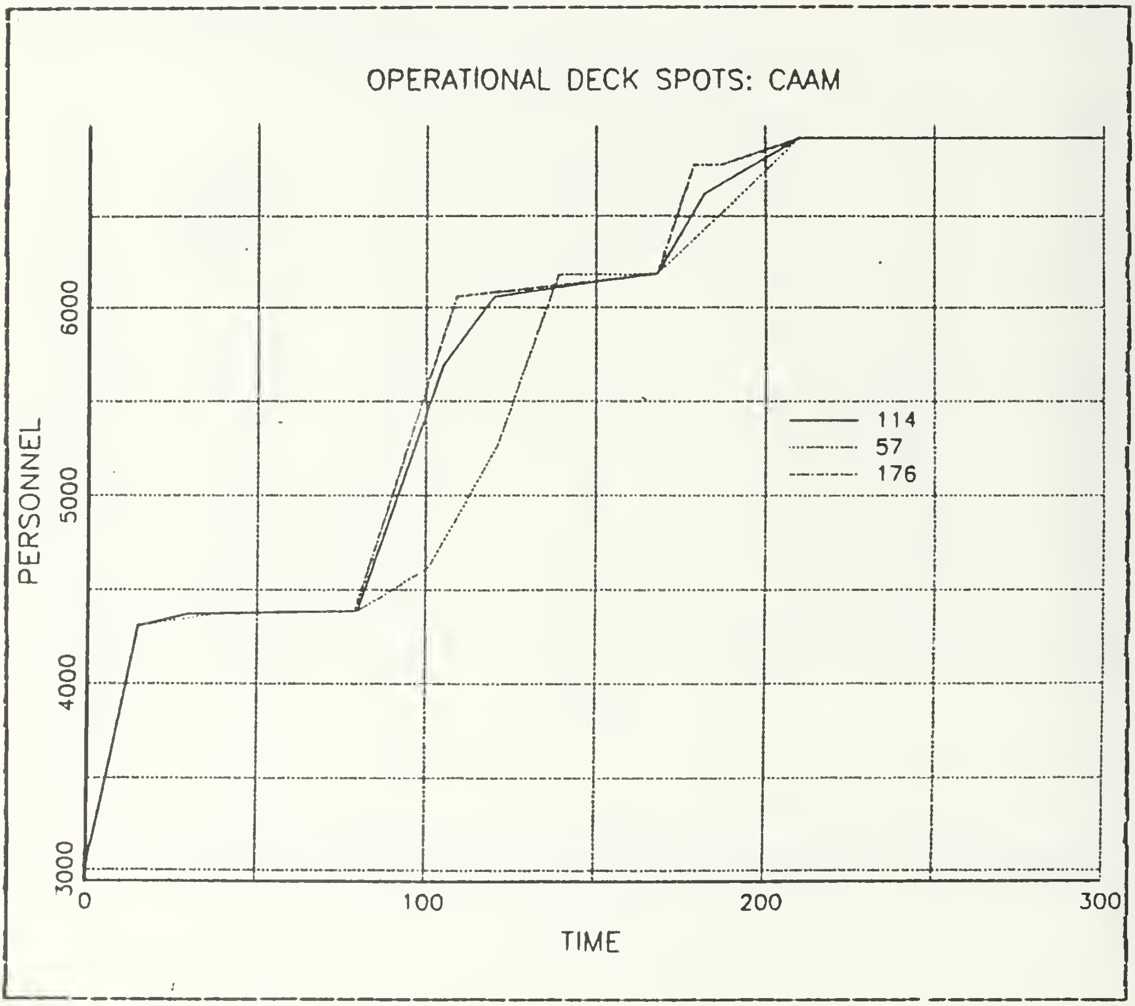


Figure 5.5 Operational Deck Spots: CAAM.

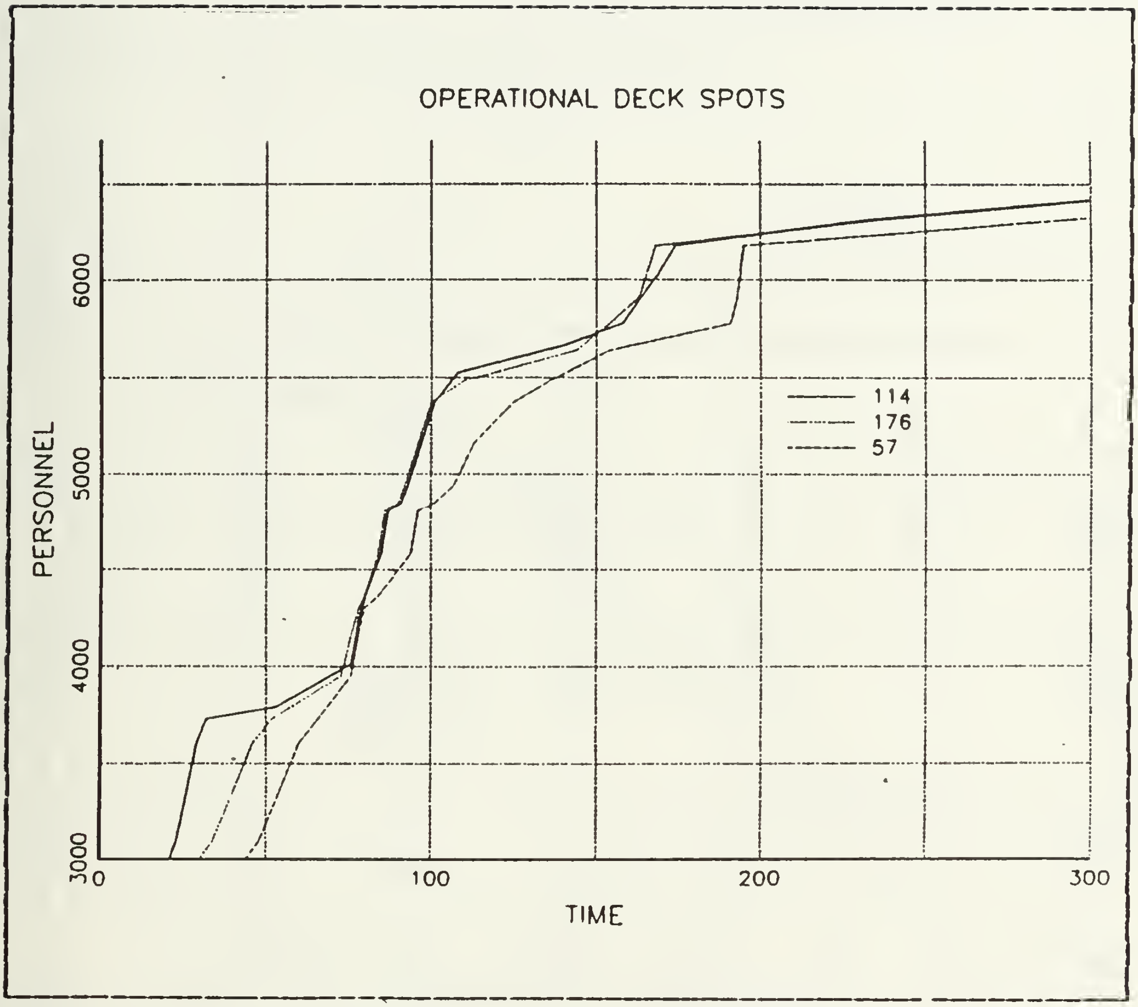


Figure 5.6 Operational Deck Spots: SHIPSHOR.

Mix of Launch Distances and Number of Helos

1. Base case: 50 NM and 156 CH-46, 80 CH-53D, 32 CH-53E
2. Alternate case #1: 35 NM and
   108 CH-46, 64 CH-53D, 32 CH-53E
3. Alternate case #2: 15 NM and
   72 CH-46, 48 CH-53D, 15 CH-53E

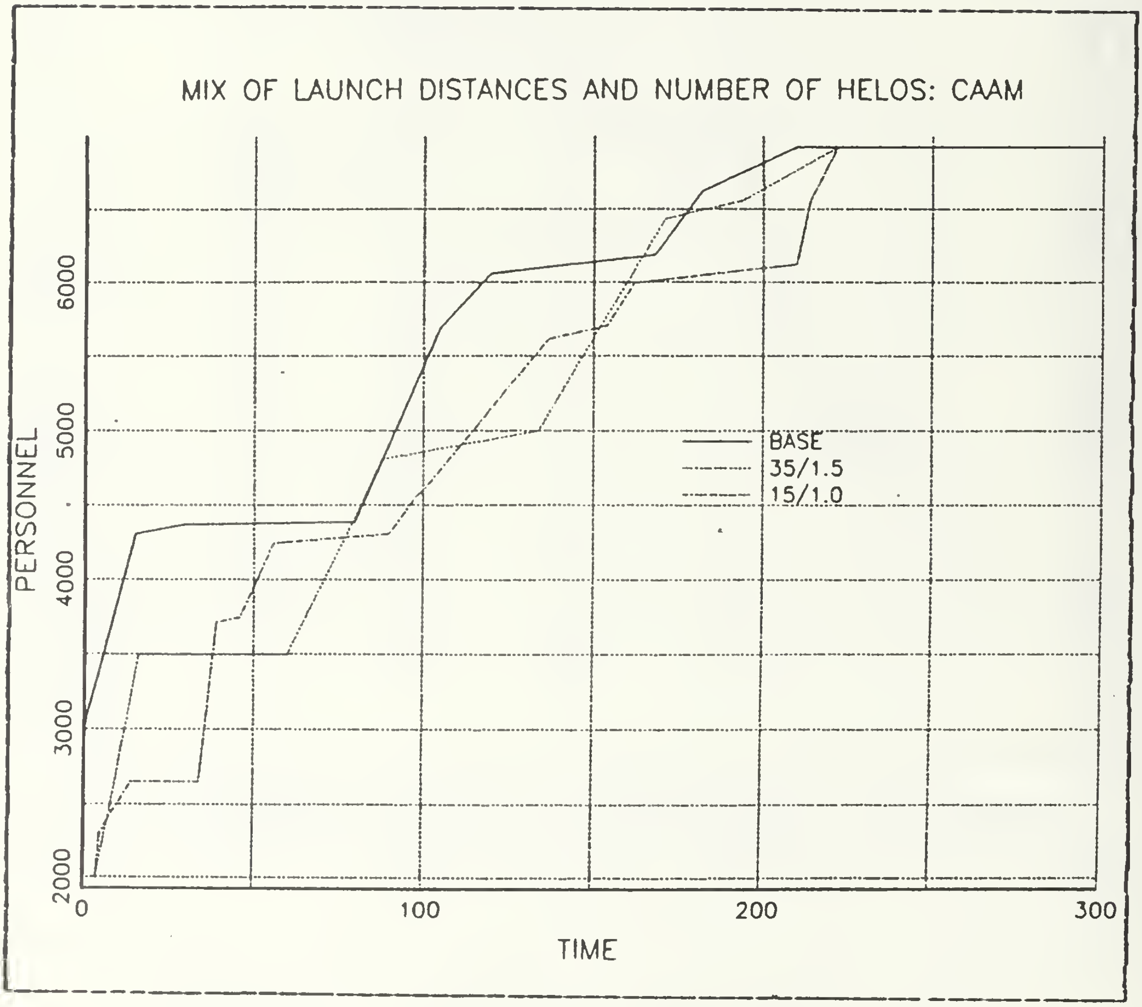


Figure 5.7 Mix of Launch Distances and # Helos: CAAM.

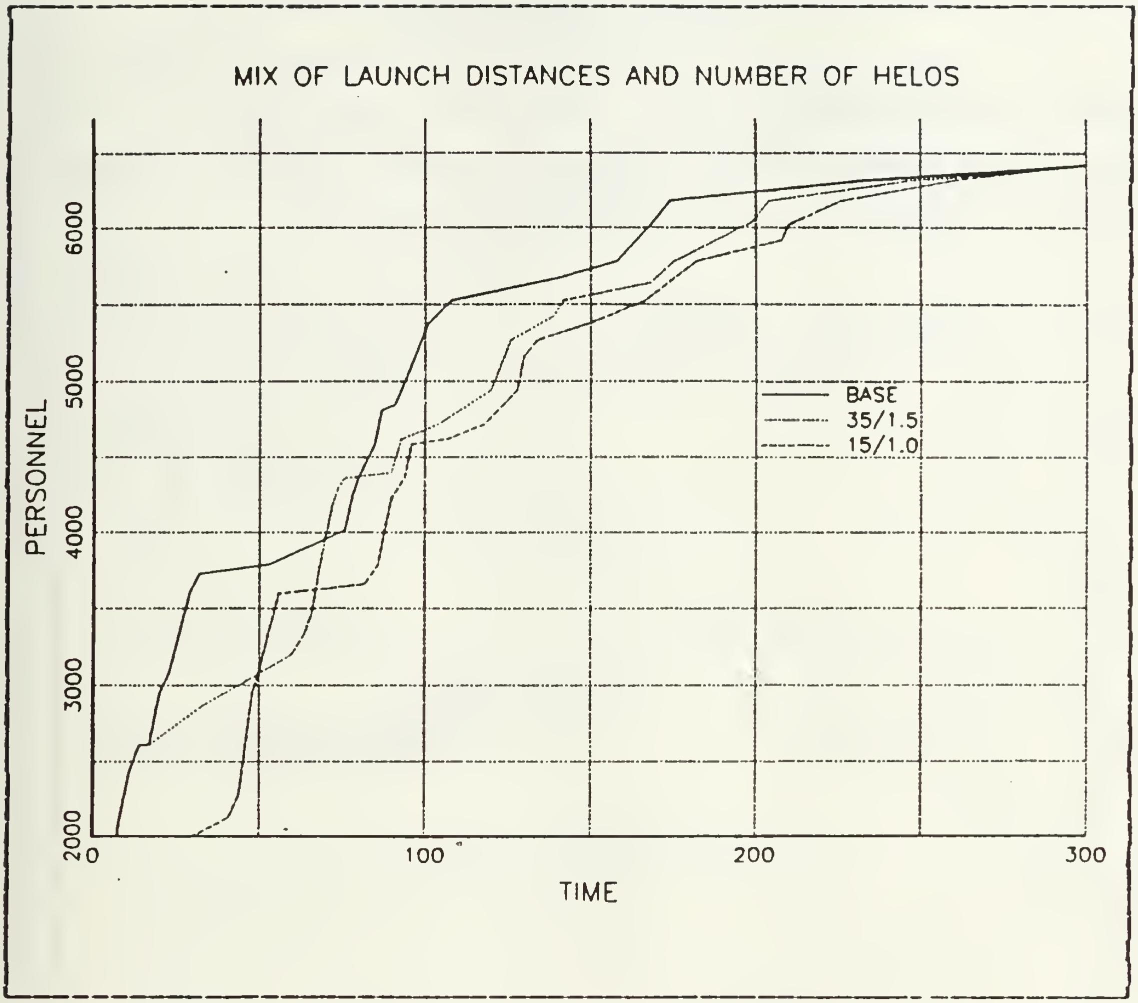


Figure 5.8 Mix of Launch Distances and # Helos: SHIPSHOR.

Although not exactly alike, CAAM and SHIPSHOR tend to exhibit similar behavior when changing parameters as shown within each graph above. Additionally, comparison between CAAM and SHIPSHOR under like conditions demonstrates that the models are producing similar landing profiles. The next section will depict the similarities and differences through direct comparison.

b. CAAM vs SHIPSHOR: Direct Comparison

The next set of graphs shows a direct comparison of CAAM and SHIPSHOR for the parameters used above. A ten percent confidence interval around the CAAM profile provides a means of judging the performance of SHIPSHOR.

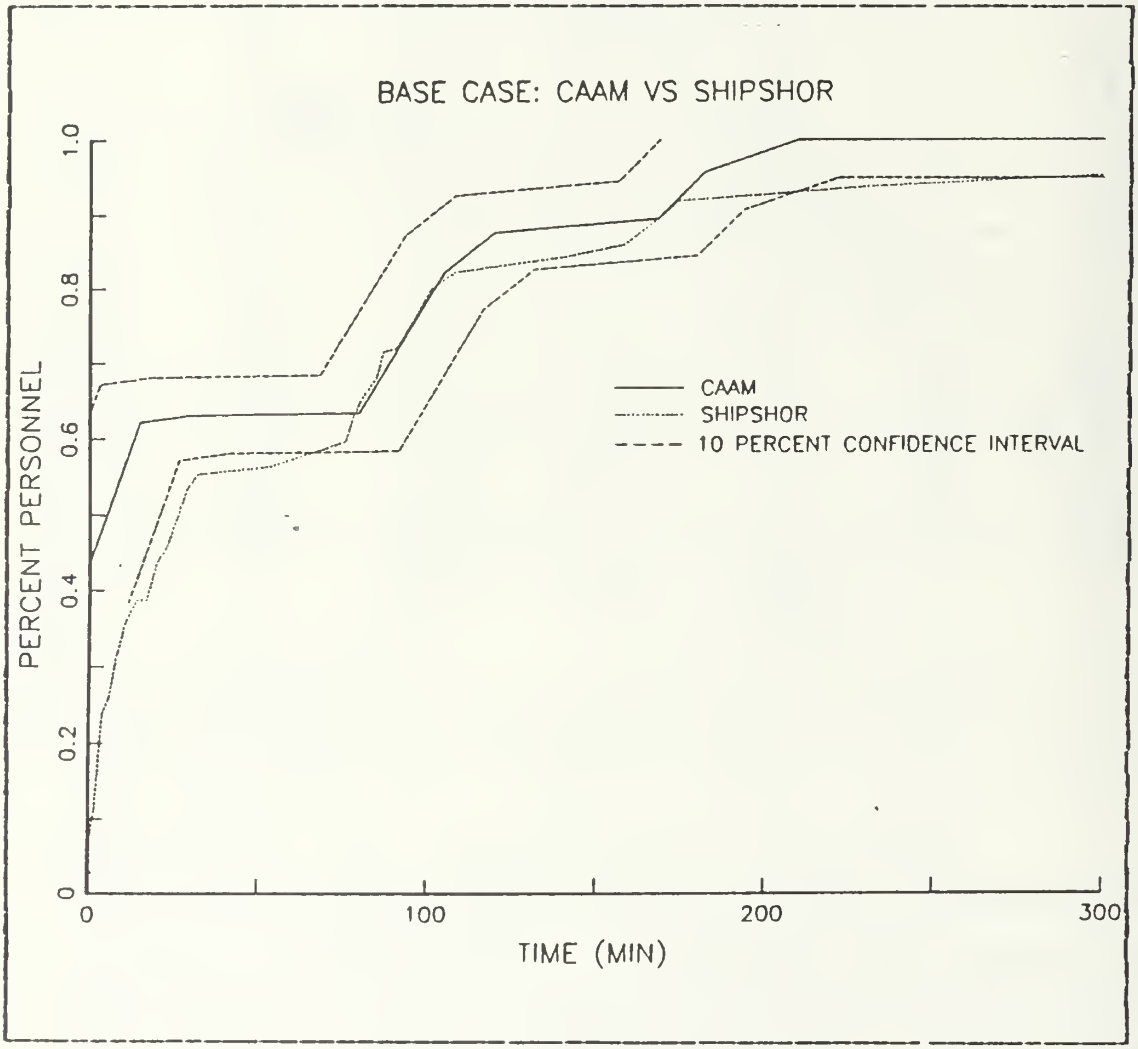
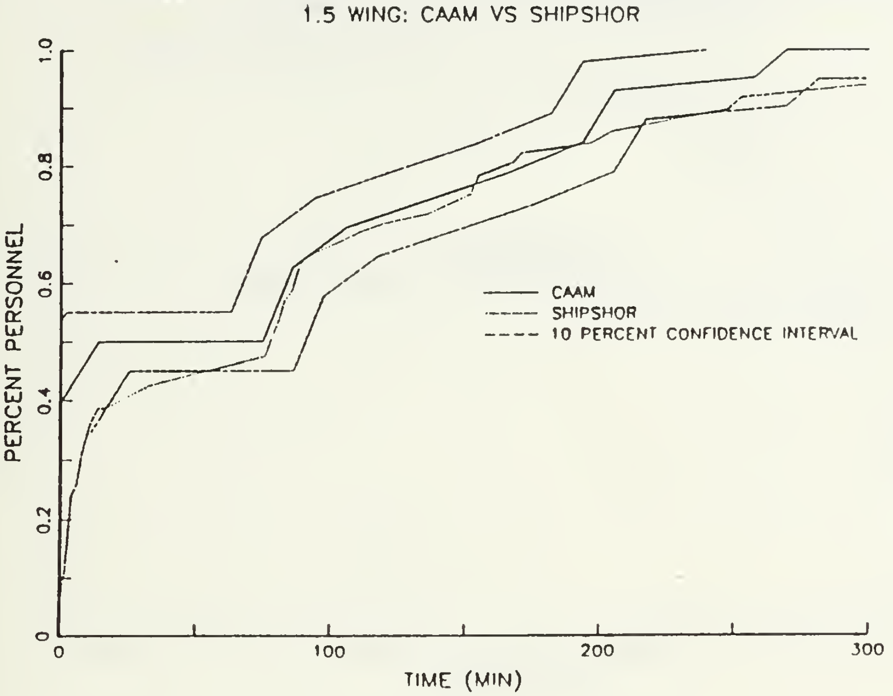


Figure 5.9 Base Case.



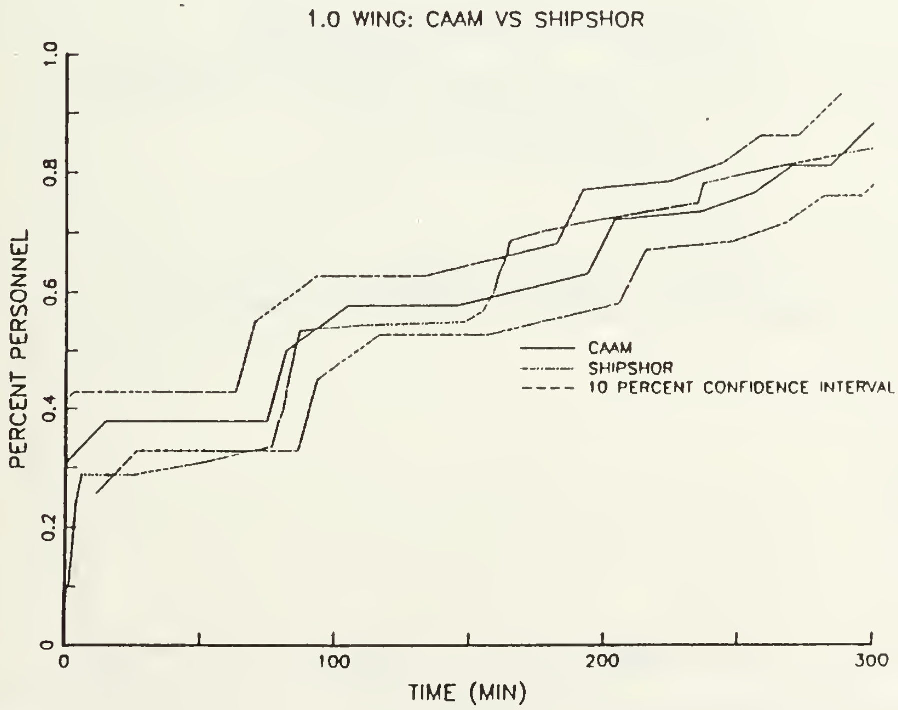


Figure 5.10 1.5 Wing and 1.0 Wing.

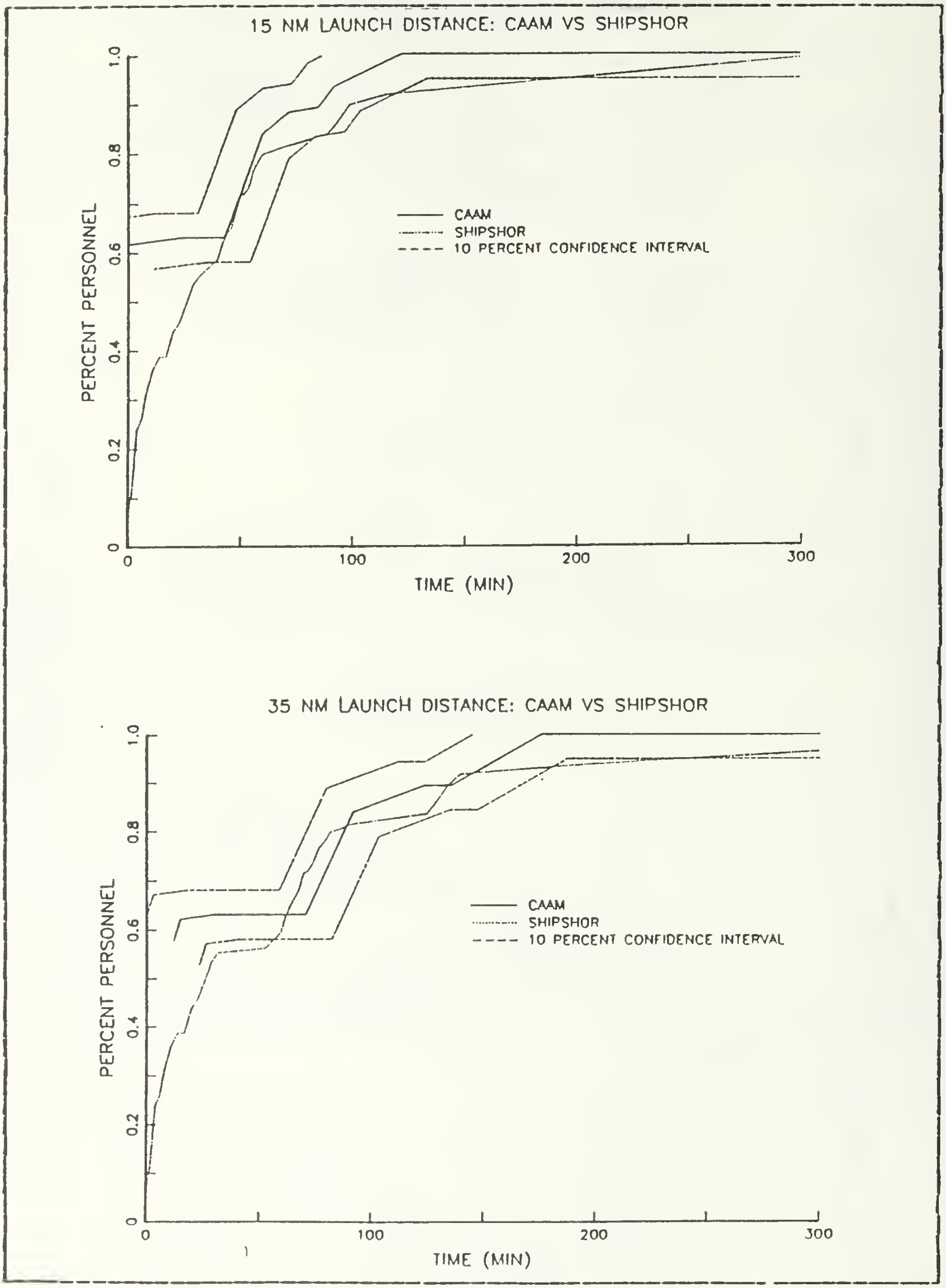


Figure 5.11 15 NM Launch Distance and 35 NM Launch Distance.

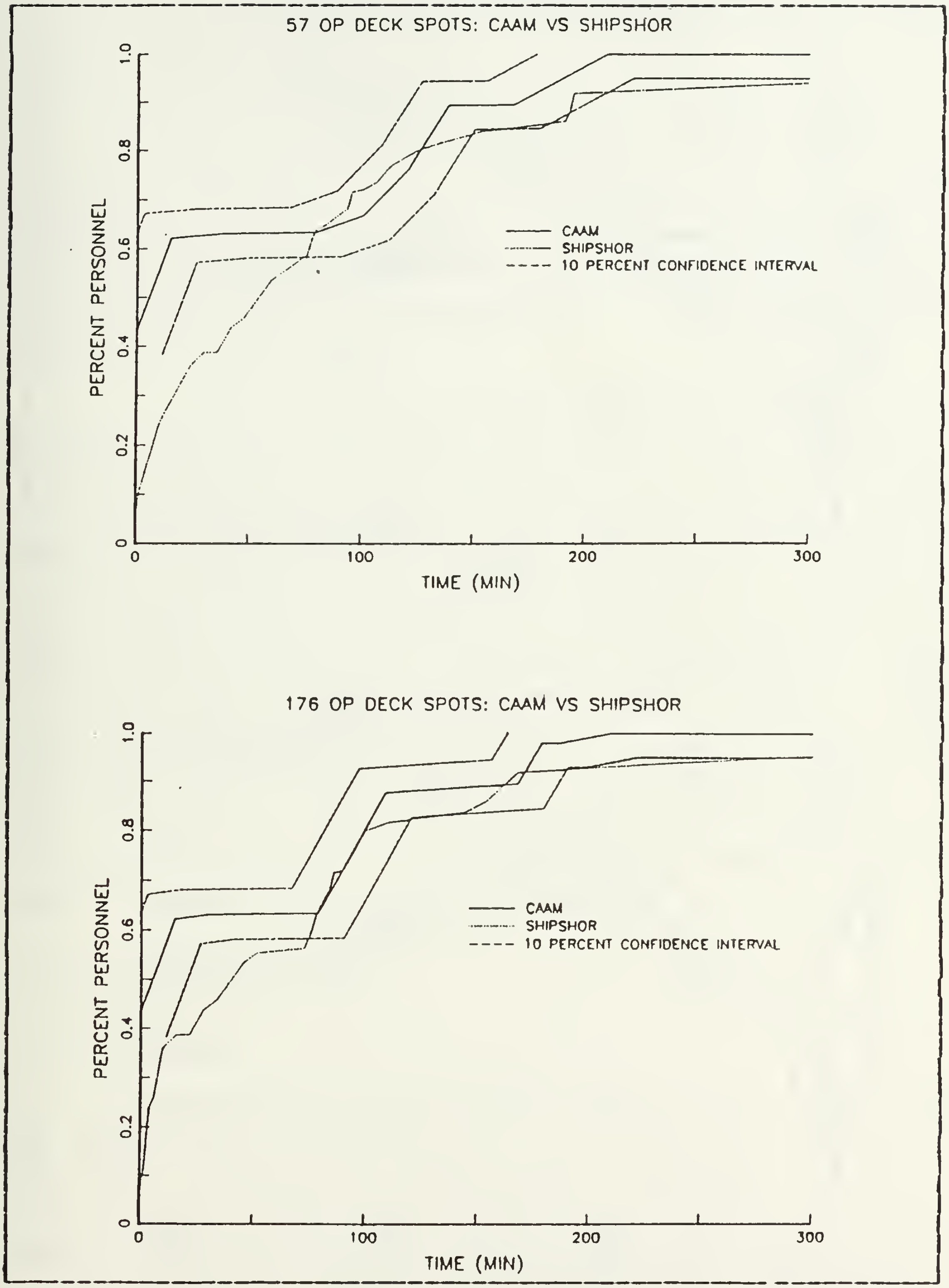


Figure 5.12 57 and 176 Operational Deck Spots.

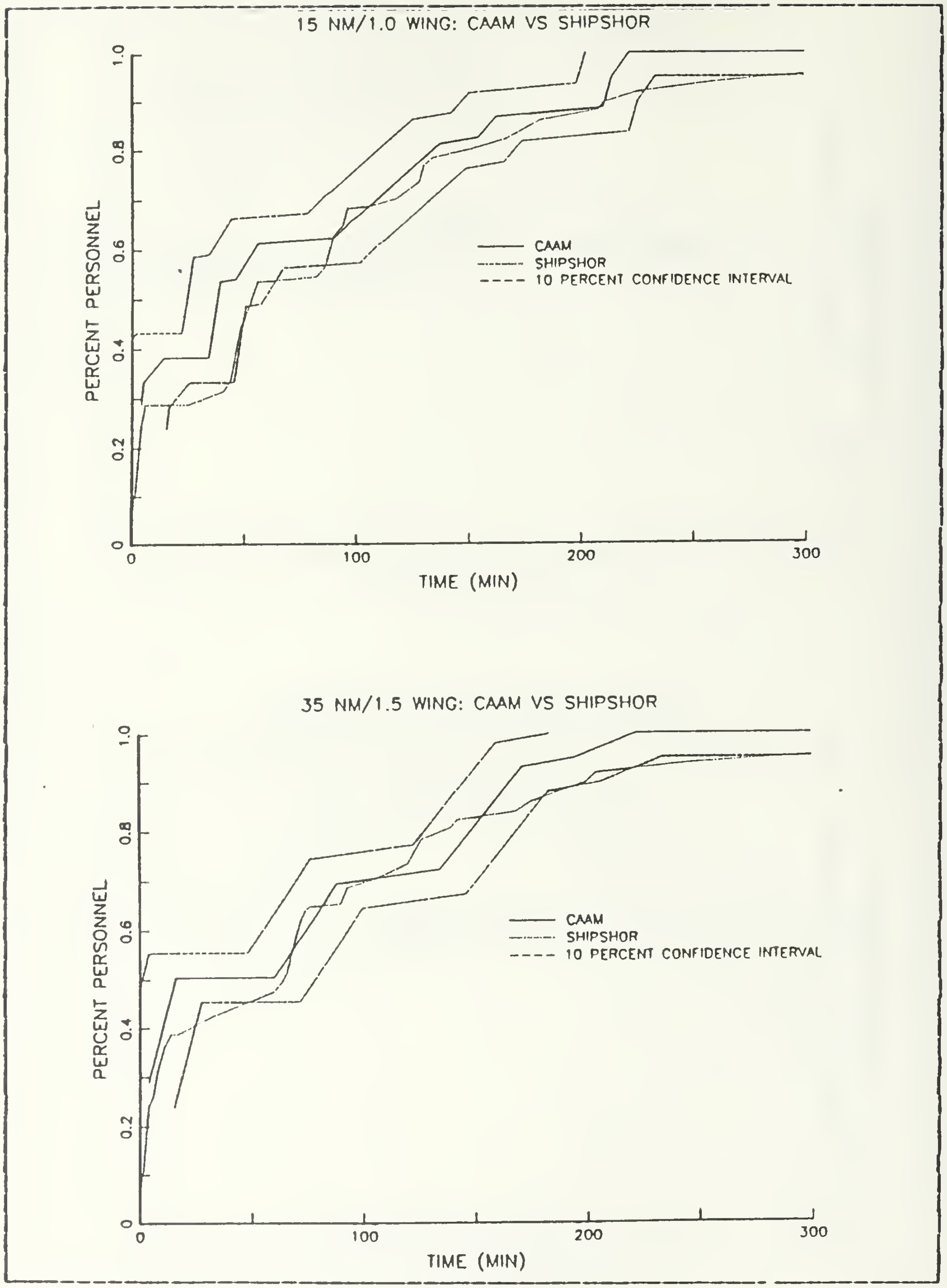


Figure 5.13 15 NM/1.0 Wing and 35 NM/1.0 Wing Distance/Number Mix.

The SHIPSHOR model, as exhibited above, tracks between the confidence limits for the most part. The biggest discrepencies between the two models is found in the first 30 or 40 minutes of the landing. CAAM displays a rapid, almost instantaneous, build-up where SHIPSHOR is more gradual. This may be explained by the manner in which SHIPSHOR increments the initial waves at one minute apart. Depending on deck spot and helicopter availibility, additional time delays may be encountered. Thus, the resulting build-up ashore is slower.

At this point one can conclude that the two models, for the particular data used, produce similar results. However, as discussed earlier in this thesis, one can not conclude that SHIPSHOR is a fully validated model. That won't happen until the model is in use and has been modified to reflect data from actual operations. There are many possible ways to disclaim the validity of this model. Although CAAM was used in a study that was published, there is nothing that guarentees the validity of CAAM. What should be recognized by the above analysis is that the confidence level in SHIPSHOR has been raised to that of CAAM. Since the authors of the study based their conclusions on the output of CAAM, there is no reason at this point to discount the output of SHIPSHOR.

C. SENSITIVITY ANALYSIS

1. General

The purpose of sensitivity analysis is to determine the sensitivity of our results to the values of the parameters used. It normally consists of a systematic varying of the parameters of interest and observing the response of the model. If the responses of interest vary greatly with slight variations or vary in a manner not expected by experience, then there may exist justification for expenditure of more time to obtain better results. [Ref. 3]

The approach of this thesis to sensitivity analysis is twofold, graphical and numerical. A graphical presentation of the landing profile with different parameters gives an overall picture of the effect of changing input values. The numerical analysis involves the selection of an MOE (measure of effectiveness). The changes observed in that MOE with a change in a particular value of input give a numerical feel for the sensitivty.

There are a myriad of MOE's for an amphibious operation. They depend on the desires of the commander, planner, or in this case, the user of the model. The commander may want to know the number of troops ashore at a given time in the operation. The planner may want to know the total time required to get a certain number of troops ashore.

For the purposes of this thesis the MOE will be the number of personnel ashore at three hours into the operation. The first several hours of a landing are most critical and from the author's experience would be of concern to a commander of the operation. The choice of a different MOE would produce different sensitivity relations in the results. Under one MOE a comparison of two sets of input parameters may be greater than under a different MOE since the profile of the landing is a step function.

The pictoral represntation will be cutoff at the 200 minute point to enable observation of the first three hours in the landing. The numerical method will introduce a term used for comparison purposes. It will take the form of:

$$\text{Sensitivity Index} = \frac{\text{Change in MOE}}{\text{Change in input parameter}}$$

$$\text{Sensitivity Index} = \frac{\text{Change in \# personnel ashore at 3 hrs.}}{\text{Change in input parameter}}$$

The method of presentation will show a set of curves reflecting the change of a single parameter from a base set of parameters. Then a table will follow with the numerical values of the sensitivity index.

Since there is a large number of parameters, there is an even greater number of combinations of parameters producing different sets of results. Therefore the following analysis will present only a small subset of all possible combinations.

2. Graphics and Numerics

The base case for the sensitivity analysis which follows is contained in Table IV . The subsequent situations list only the parameters of interest, then the graph of the different cases, followed by the numerics.

TABLE IV

Input Data for the Base Case

Helo Launch Dist	Number of Helicopters			Attrition
	CH-46	CH-53D	CH-53E	
10 NM	130	63	27	.8

Number of boats

LVT	LCM-6	LCM-8	LCU	LST	LARC
249	52	42	28	0	12

Speed (type load)		Operatonal Deck Spots	Number of LZ's
Internal	External		
120 kts	100 kts	100	2

Beach to LZ Distance	Size of Landing Zone	LVT Launch distance
5 NM	48 helos	5 NM

Number of beaches	Beach Size
one	6 LVTs

Helicopter Time Factors

1. Base case: Helicopter time factors table
2. Alternate #1: Load time 6 minutes
3. Alternate #2: Load time 10 minutes
4. Alternate #3: Load time 15 minutes
5. Alternate #4: Double helo time factors table

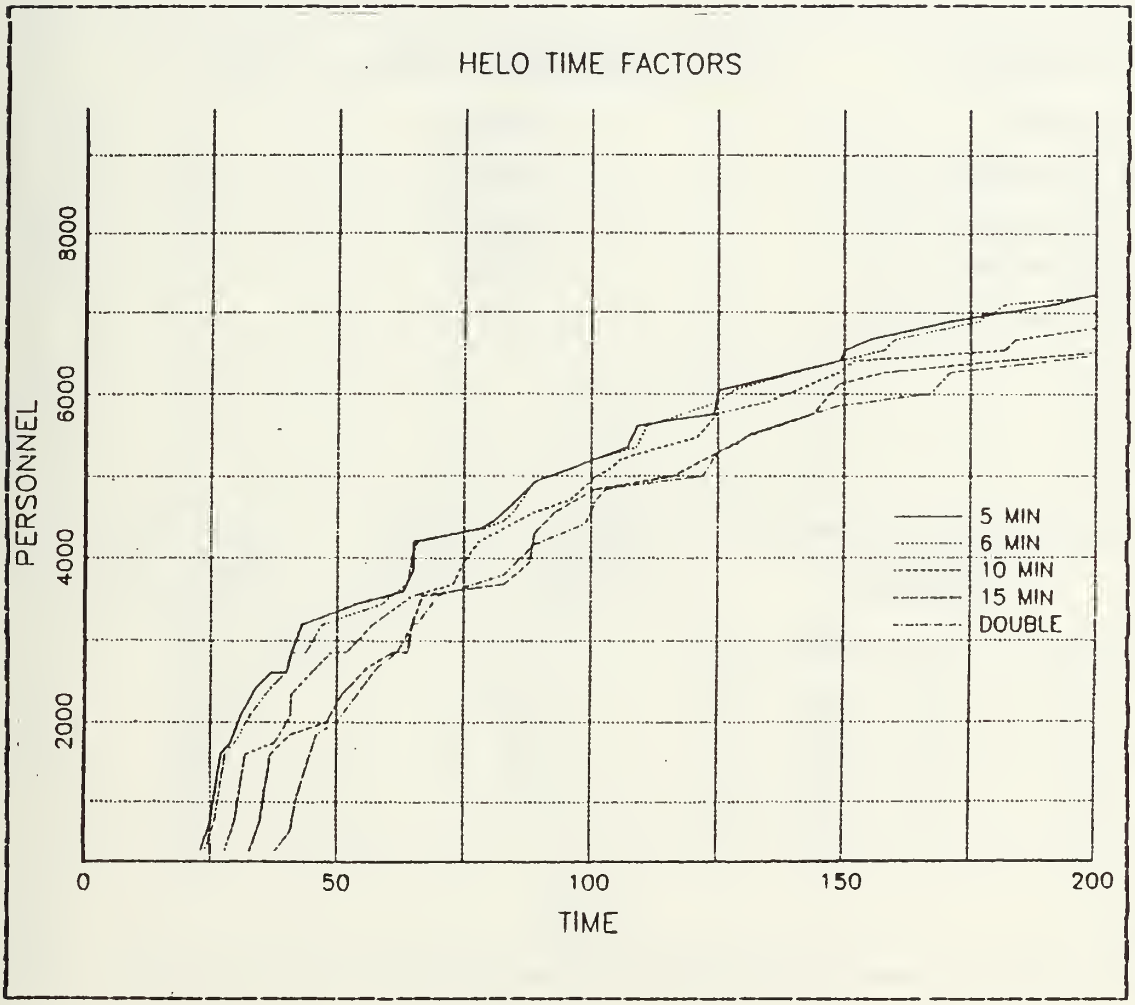


Figure 5.14 Helicopter Time Factors.

TABLE V

Sensitivity Index: Helicopter Time Factors

Parameter	Number personnel ashore (3 hrs.)
Base:	6894
Alternate #1:	7110
Alternate #2:	6542
Alternate #3:	6404
Alternate #4:	6266

Change in Parameter (in minutes)		Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Base - Alt #2:	5	352	-71
Alt #1 - Alt #2:	4	568	-142
Alt #2 - Alt #3:	5	138	-28
Base - Alt #3:	10	490	-49

Beach size

1. Base case: 6 LVTs
2. Alternate #1: 2 LVTs
3. Alternate #2: 4 LVTs
4. Alternate #3: 8 LVTs
5. Alternate #4: 10 LVTs

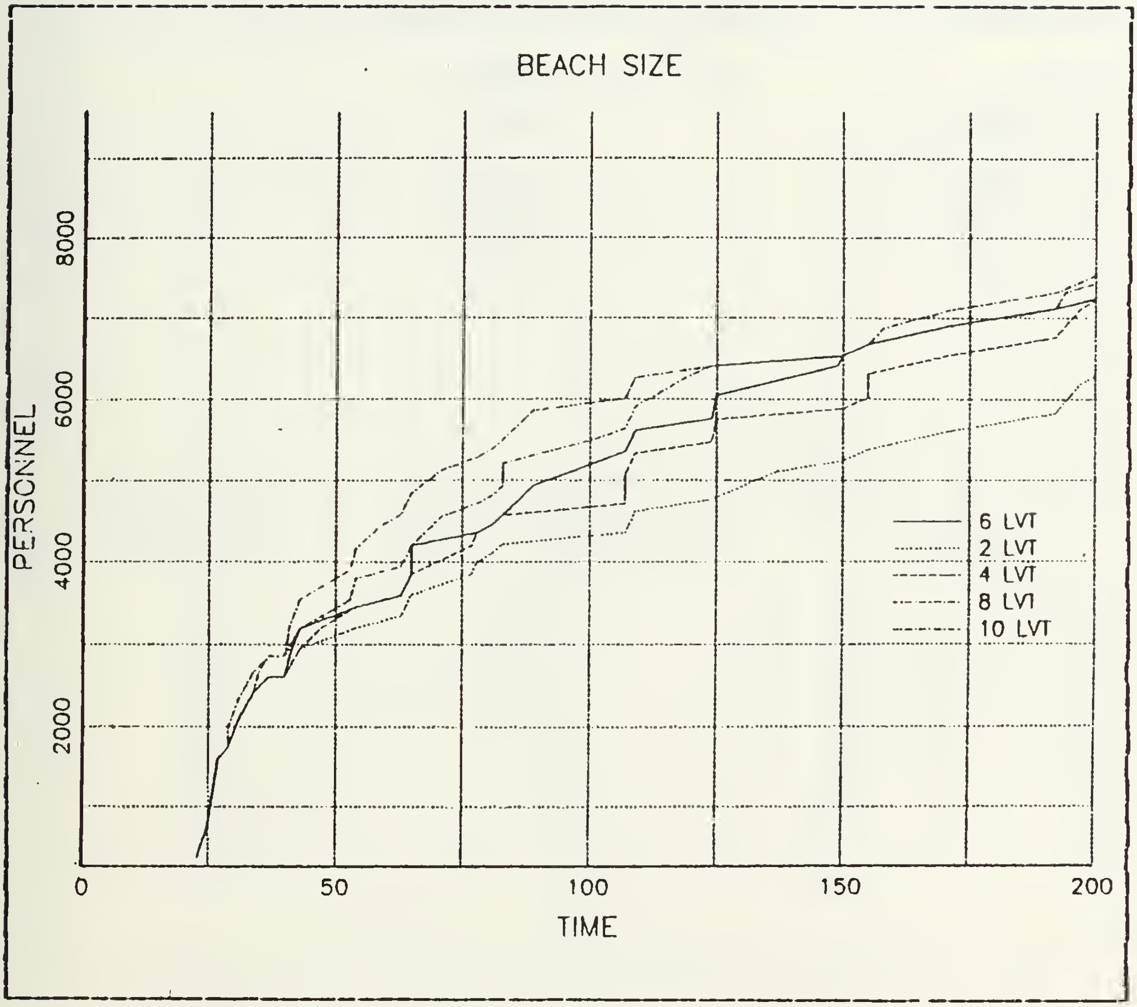


Figure 5.15 Beach Size.

TABLE VI

Sensitivity Index: Beach Size

Parameter	Number personnel ashore (3 hrs.)
Base:	6894
Alternate #1:	5604
Alternate #2:	6534
Alternate #3:	6894
Alternate #4:	7097

Change in Parameter (in LVTs)	Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Base - Alt #4: 2	360	+180
Alt #1 - Alt #2: 2	930	+465
Alt #2 - Alt #4: 6	563	+94
Alt #1 - Alt #4: 8	1493	+187
Alt #1 - Alt #3: 6	1290	+215

Landing Zone Size

1. Base case: 48 CH-46 Helicopters
2. Alternate #1: 6 CH-46 Helicopters
3. Alternate #2: 4 CH-46 Helicopters
4. Alternate #3: 3 CH-46 Helicopters
5. Alternate #4: 2 CH-46 Helicopters

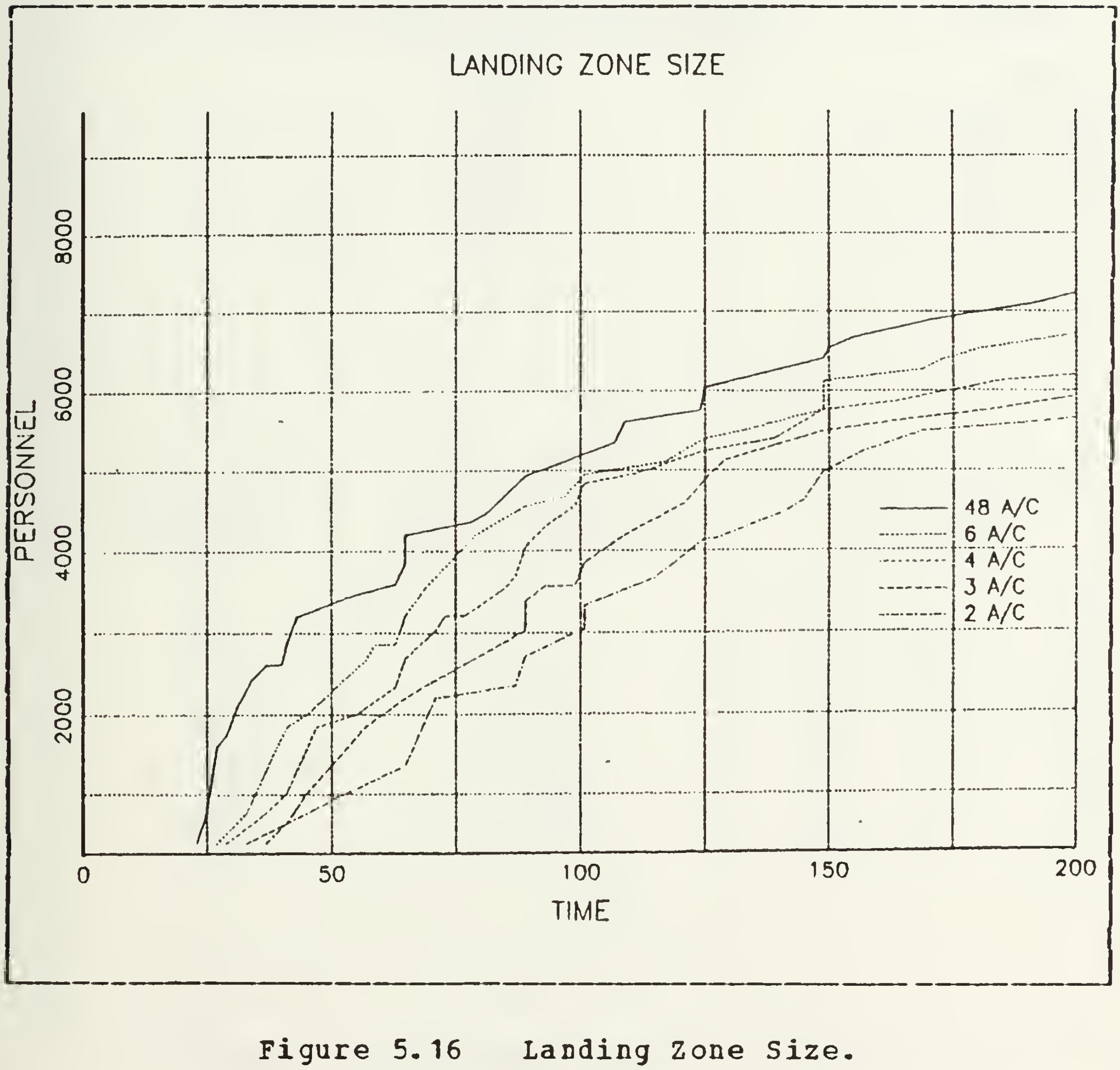


Figure 5.16 Landing Zone Size.

TABLE VII

Sensitivity Index: Landing Zone Size

Parameter	Number personnel ashore (3 hrs.)
Base:	6894
Alternate #1:	6532
Alternate #2:	5866
Alternate #3:	5650
Alternate #4:	5495

Change in Parameter (in CH-46s)	Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Base - Alt #1: 42	362	-9
Alt #1 - Alt #2: 2	666	-333
Alt #2 - Alt #3: 1	216	-216
Alt #3 - Alt #4: 1	155	-155
Alt #1 - Alt #3: 3	882	-294
Alt #1 - Alt #4: 4	1037	-260
Alt #2 - Alt #4: 2	371	-186

Operational Deck Spots

1. Base case: 100
2. Alternate #1: 176
3. Alternate #2: 114
4. Alternate #3: 60
5. Alternate #4: 20

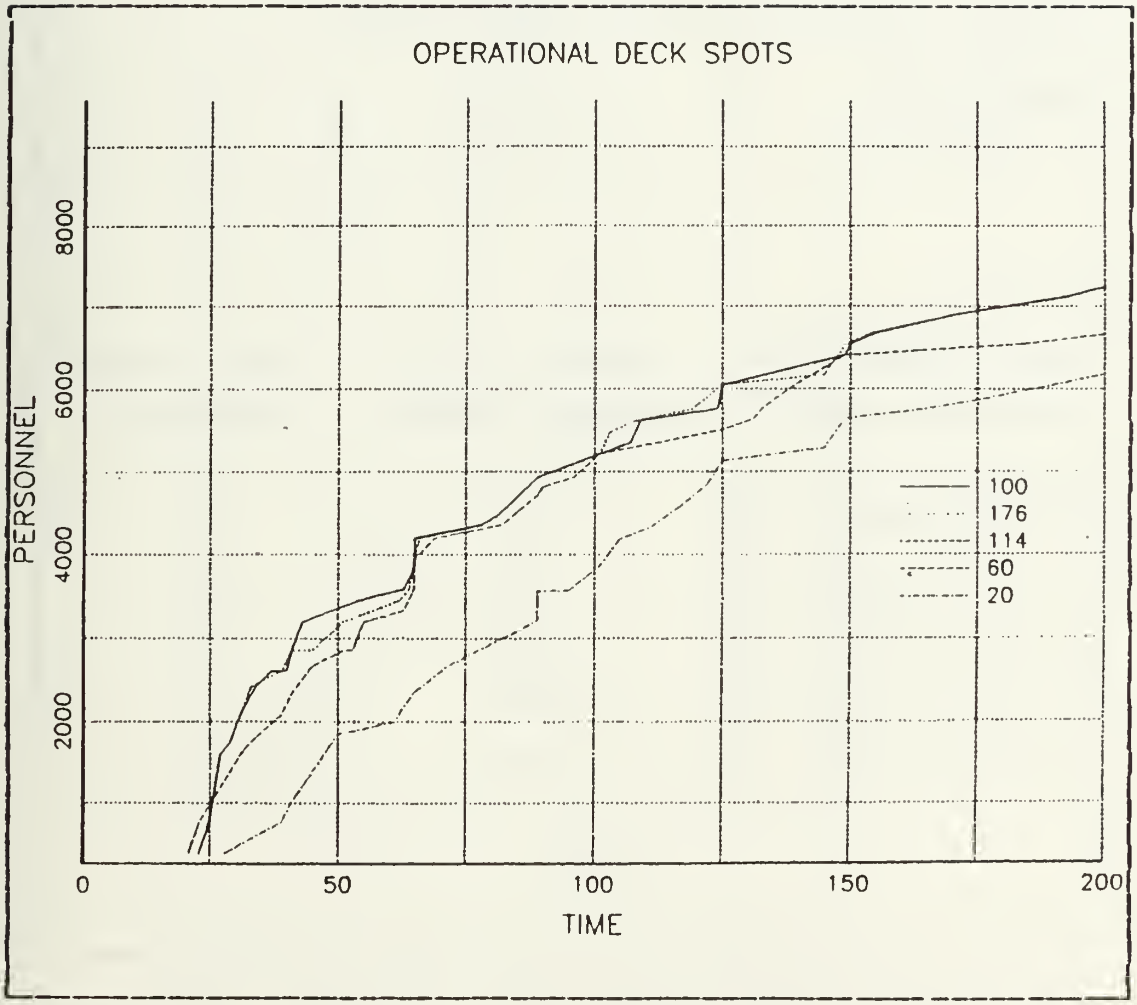


Figure 5.17 Operational Deck Spots.

TABLE VIII

Sensitivity Index: Operational Deck Spots

Parameter	Number personnel ashore (3 hrs.)
Base:	6894
Alternate #1:	6894
Alternate #2:	6894
Alternate #3:	6404
Alternate #4:	5866

Change in Parameter (in deck spots)		Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Base - Alt #1:	76	0	0
Alt #1 - Alt #2:	62	0	0
Alt #2 - Alt #3:	54	490	-10
Alt #3 - Alt #4:	40	538	-11
Alt #1 - Alt #3:	116	882	-7
Alt #1 - Alt #4:	156	1037	-6
Alt #2 - Alt #4:	94	371	-4

Number of 6 Helicopter Sized Landing Zones

1. Case #1: 6 helicopter size / 1 landing zones
2. Case #2: 6 helicopter size / 2 landing zones
3. Case #3: 6 helicopter size / 3 landing zones
4. Case #4: 6 helicopter size / 4 landing zones

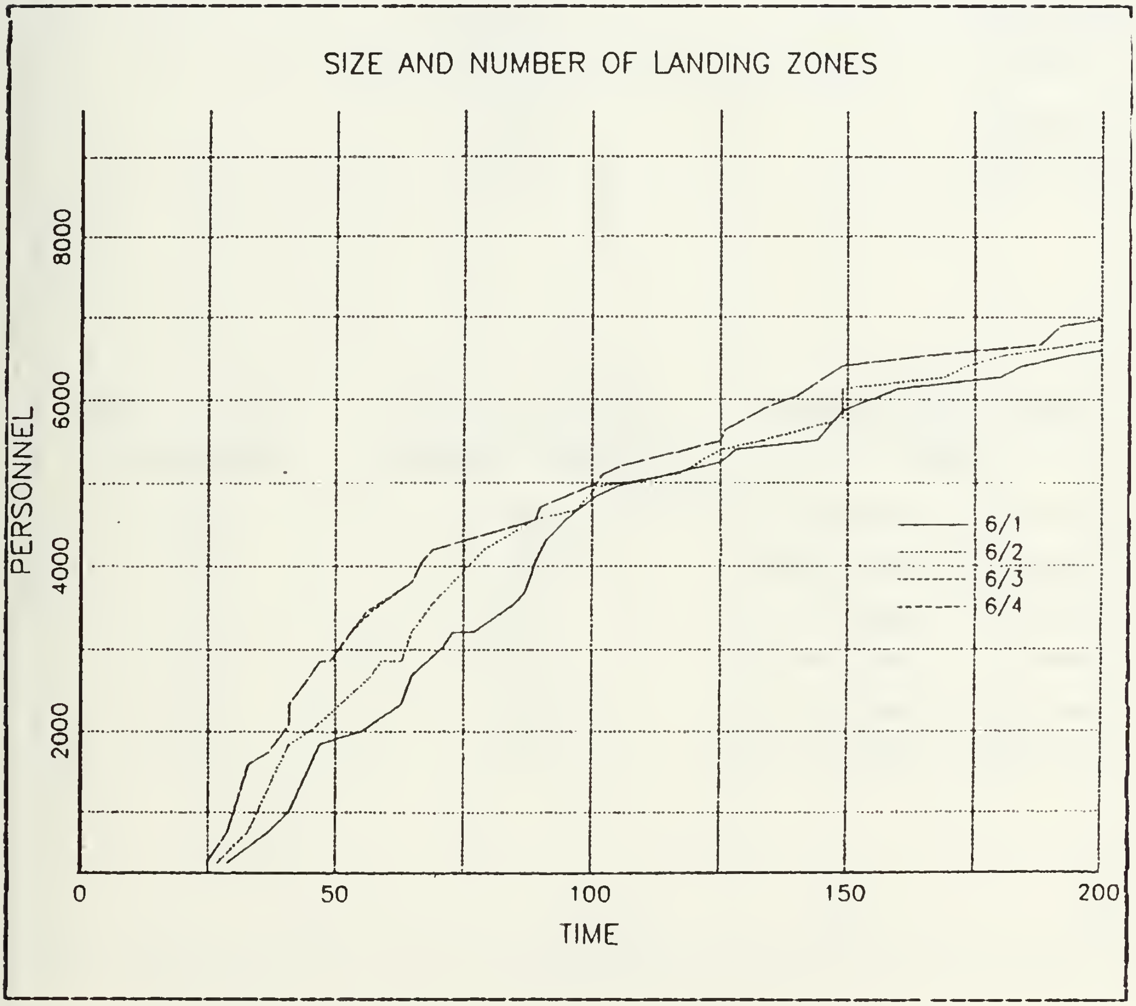


Figure 5.18 Number of 6 Helicopter Sized Landing Zones.

TABLE IX

Sensitivity Index: # of 6 Helo Sized LZs

Parameter	Number personnel ashore (3 hrs.)
Case #1:	6266
Case #2:	6394
Case #3:	6532
Case #4:	6532

Change in Parameter (in # LZs)	Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Case #1 - Case #2: 1	128	+128
Case #2 - Case #3: 1	138	+138
Case #3 - Case #4: 1	0	0
Case #1 - Case #3: 2	266	+133
Case #1 - Case #4: 3	266	+89
Case #2 - Case #4: 2	138	+69

Number of 4 Helicopter Sized Landing Zones

1. Case #1: 4 helicopter size / 1 landing zones
2. Case #2: 4 helicopter size / 2 landing zones
3. Case #3: 4 helicopter size / 3 landing zones
4. Case #4: 4 helicopter size / 4 landing zones

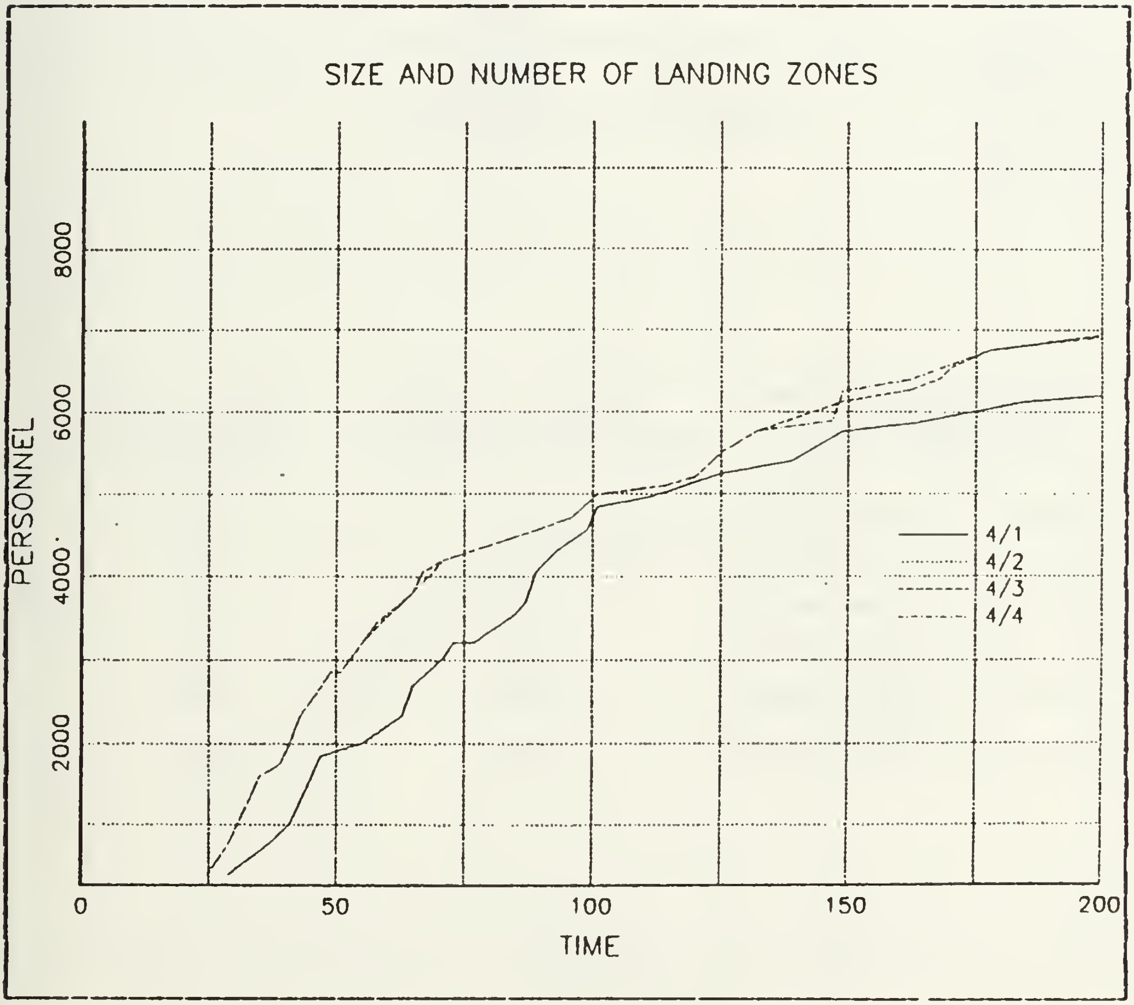


Figure 5.19 Number of 4 Helicopter Sized Landing Zones.

TABLE X

Sensitivity Index: # of 4 Helo Sized LZs

Parameter	Number personnel ashore (3 hrs.)
Case #1:	5866
Case #2:	5866
Case #3:	6756
Case #4:	6756

Change in Parameter (in # LZs)	Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Case #1 - Case #2: 1	0	0
Case #2 - Case #3: 1	890	+890
Case #3 - Case #4: 1	0	0
Case #1 - Case #3: 2	890	+445
Case #1 - Case #4: 3	890	+297
Case #2 - Case #4: 2	890	+445

Helicopter Launch Distance

1. Base case: 10 NM
2. Alternate #1: 20 NM
3. Alternate #2: 30 NM
4. Alternate #3: 40 NM
5. Alternate #4: 50 NM

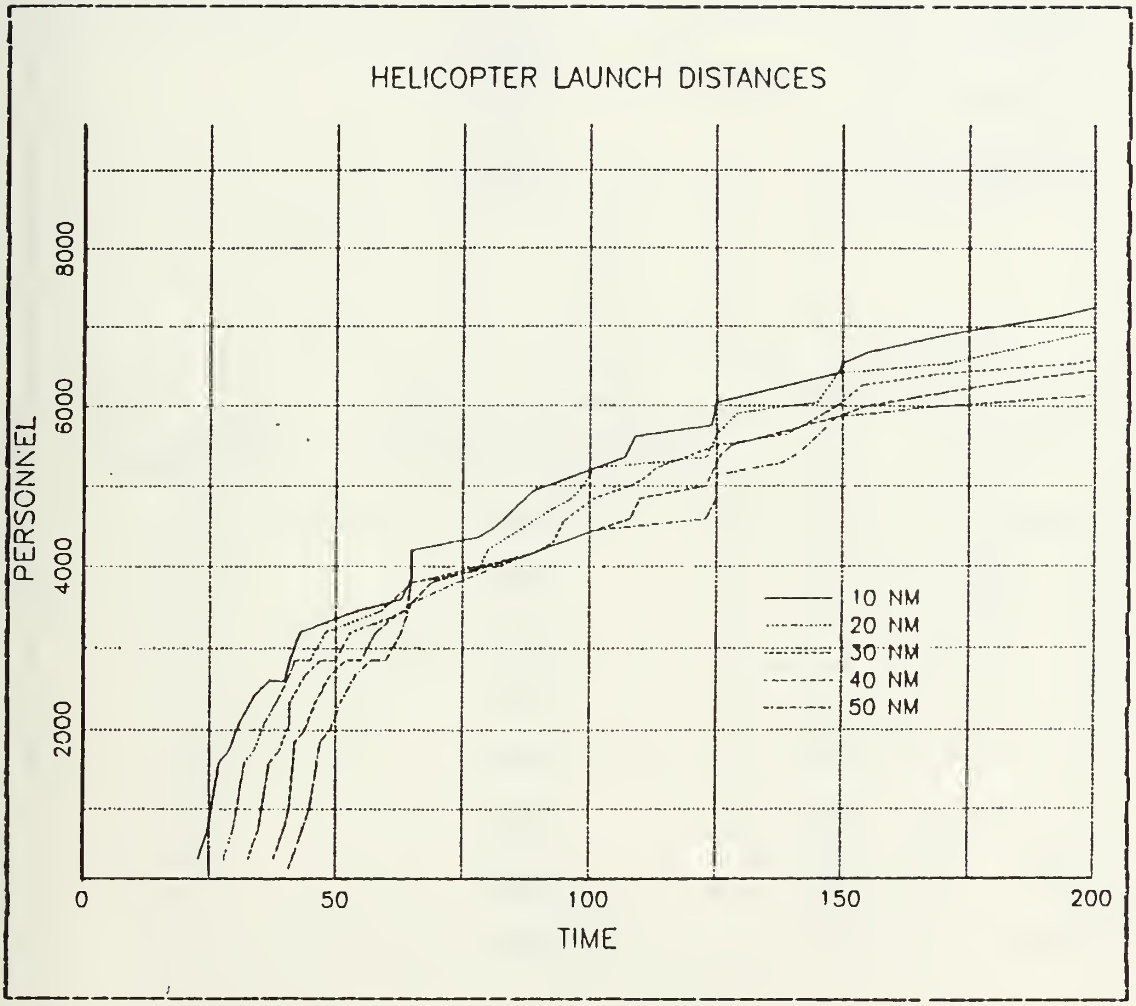


Figure 5.20 Helicopter Launch Distance.

TABLE XI

Sensitivity Index: Helicopter Launch Distance

Parameter	Number personnel ashore (3 hrs.)
Base:	6894
Alternate #1:	6532
Alternate #2:	6404
Alternate #3:	6266
Alternate #4:	6004

Change in Parameter (in NM)		Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Base - Alt #1:	10	362	-37
Alt #1 - Alt #2:	10	128	-13
Alt #2 - Alt #3:	10	138	-14
Alt #3 - Alt #4:	40	538	-11
Alt #1 - Alt #3:	20	266	-14
Alt #1 - Alt #4:	30	528	-18
Alt #2 - Alt #4:	20	400	-20
Base - Alt #2:	20	490	-25
Base - Alt #3:	30	628	-21
Base - Alt #4:	40	890	-23

Number of Helicopters

1. Base case: 130 CH-46, 63 CH-53D, 27 CH-53E
2. Alternate #1: 156 CH-46, 80 CH-53D, 32 CH-53E
3. Alternate #2: 108 CH-46, 64 CH-53D, 32 CH-53E
4. Alternate #3: 80 CH-46, 40 CH-53D, 15 CH-53E
5. Alternate #4: 72 CH-46, 48 CH-53D, 16 CH-53E

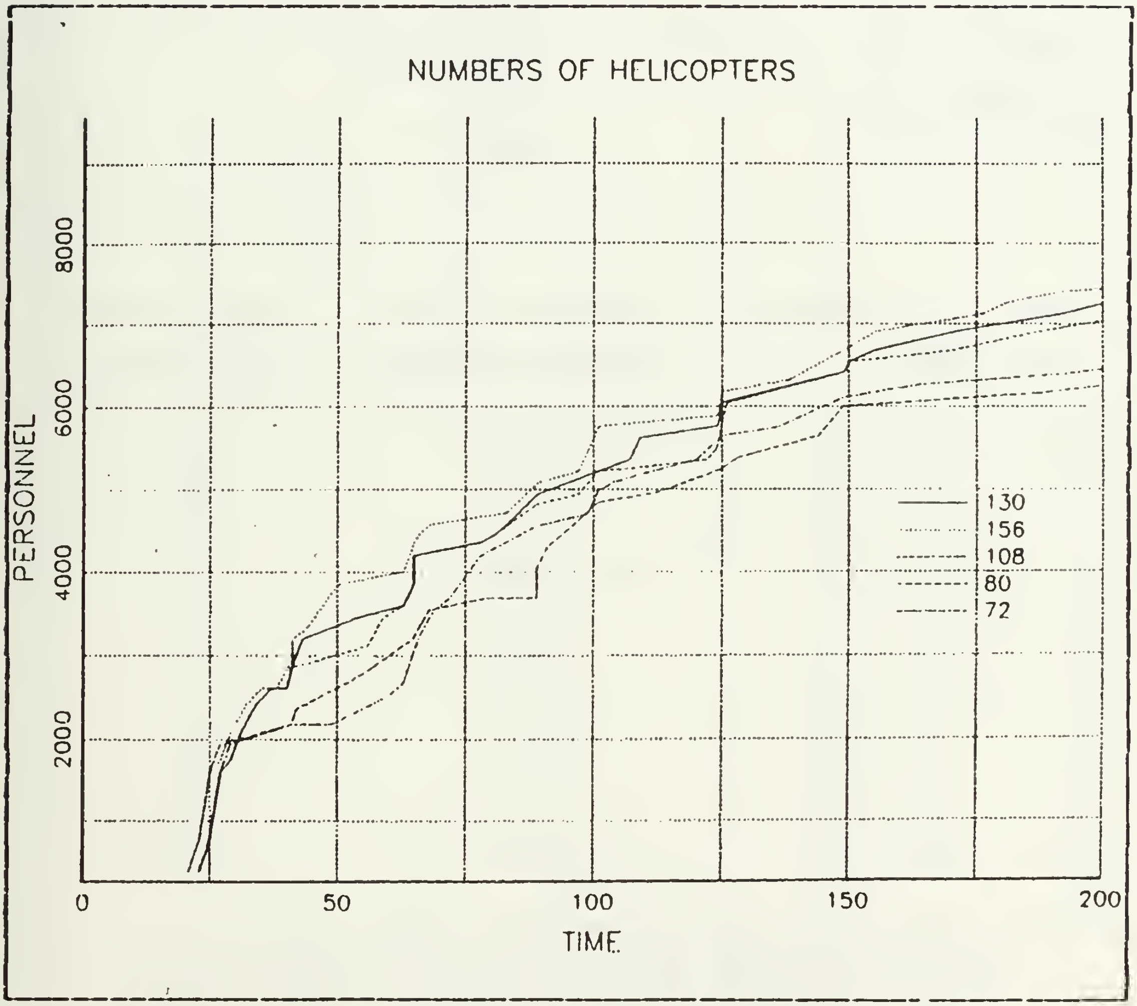


Figure 5.21 Number of Helicopters.

TABLE XII

Sensitivity Index: Number of Helicopters

Parameter	Number personnel ashore (3 hrs.)
Base:	6894
Alternate #1:	7110
Alternate #2:	6670
Alternate #3:	6007
Alternate #4:	6266

Change in Parameter (no unit)	Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Base - Alt #1:	+216	N/A
Alt #1 - Alt #2:	-224	N/A
Alt #2 - Alt #3:	-663	N/A
Alt #3 - Alt #4:	+259	N/A
Alt #1 - Alt #3:	-1103	N/A
Alt #1 - Alt #4:	-844	N/A
Alt #2 - Alt #4:	-404	N/A
Base - Alt #2:	-224	N/A
Base - Alt #3:	-887	N/A
Base - Alt #4:	-628	N/A

Note: Since there were three parameters changed at one time during this run, no unit could be devised for the denominator of the sensitivity index. However, an idea of the model's response can be obtained from the change in MOE values.

Number of CH-46 Helicopters

1. Base case: 130
2. Alternate #1: 108
3. Alternate #2: 72
4. Alternate #3: 54
5. Alternate #4: 36

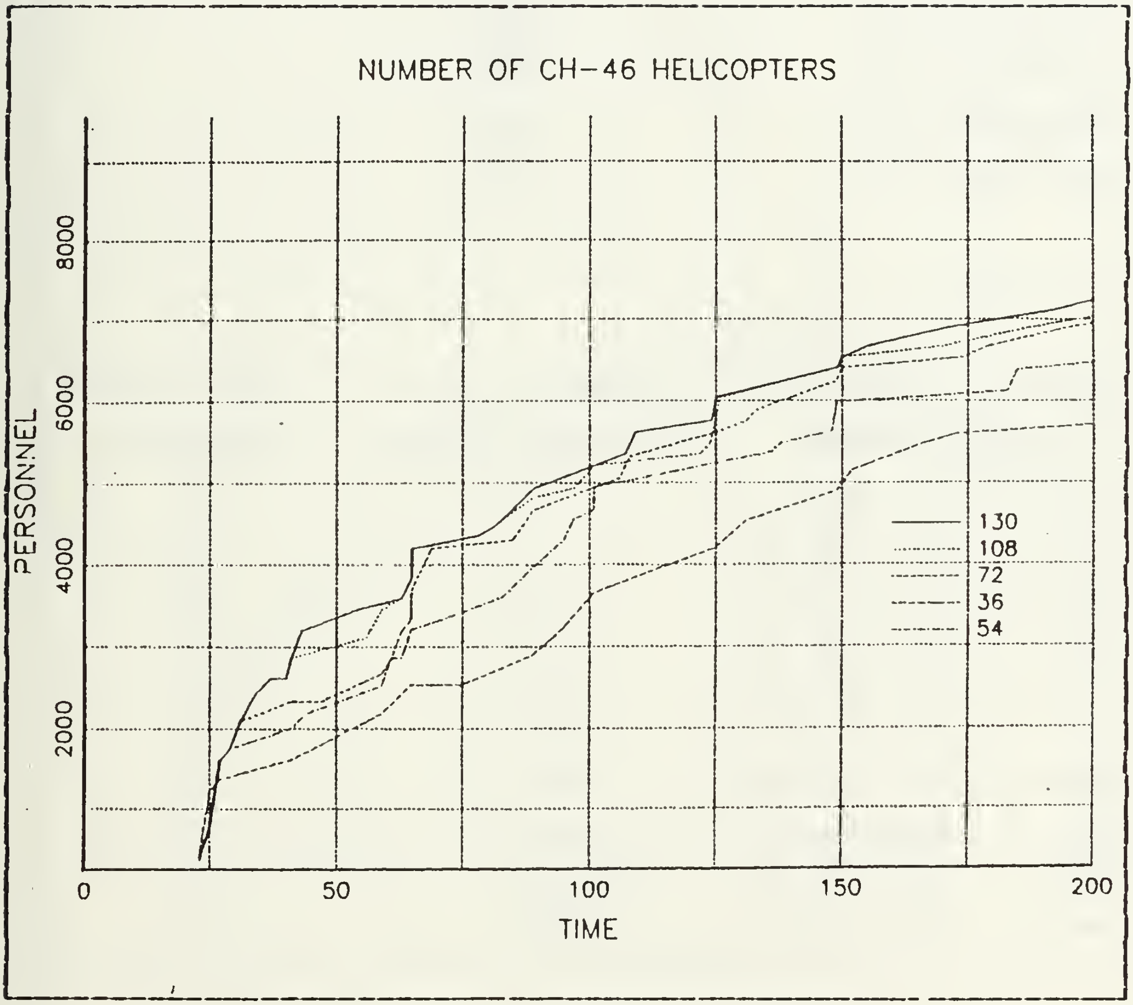


Figure 5.22 Number of CH-46 Helicopters.

TABLE XIII

Sensitivity Index: Number of CH-46 Helicopters

Parameter	Number personnel ashore (3 hrs.)
Base:	6894
Alternate #1:	6670
Alternate #2:	6670
Alternate #3:	5987
Alternate #4:	5599

Change in Parameter (in # of CH-46s)	Change in MOE (# troops ashore)	Sens. Index (MOE/para.).
Base - Alt #1: 22	224	-11
Alt #1 - Alt #2: 36	0	0
Alt #2 - Alt #3: 18	683	-38
Alt #3 - Alt #4: 18	298	-17
Alt #1 - Alt #3: 54	683	-13
Alt #1 - Alt #4: 72	1071	-15
Alt #2 - Alt #4: 36	1071	-30
Base - Alt #2: 58	224	-4
Base - Alt #3: 76	907	-12
Base - Alt #4: 94	1295	-14

Number of Beaches

1. Base case: 1
2. Alternate #1: 2
3. Alternate #2: 3
4. Alternate #3: 4

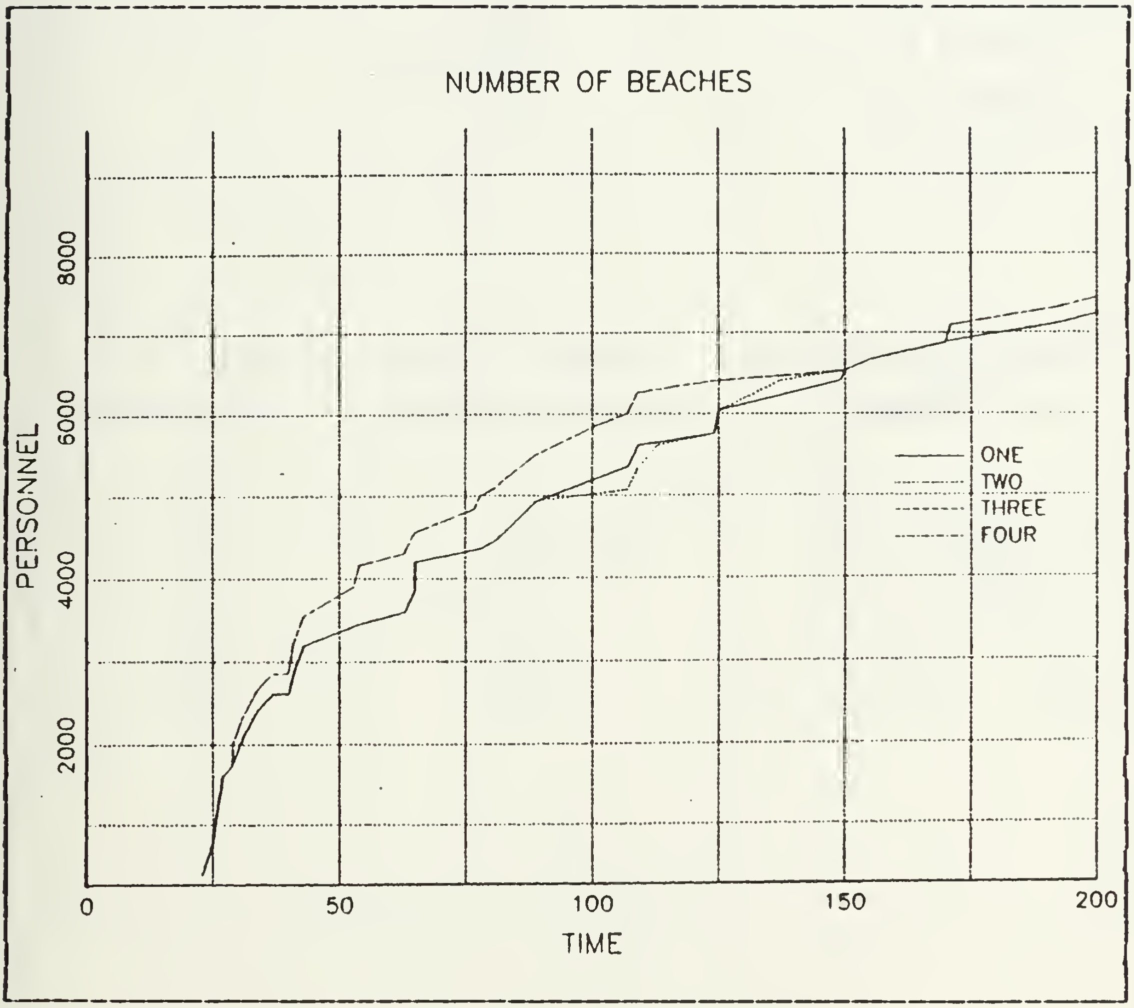


Figure 5.23 Number of Beaches.

TABLE XIV
Sensitivity Index: Number of Beaches

Parameter	Number personnel ashore (3 hrs.)
Base:	6894
Alternate #1:	6894
Alternate #2:	7097
Alternate #3:	7097

Change in Parameter (in beaches)		Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Base - Alt #1:	1	0	0
Alt #1 - Alt #2:	1	203	+203
Alt #2 - Alt #3:	1	0	0
Alt #1 - Alt #3:	2	203	+102
Base - Alt #2:	2	203	+102
Base - Alt #3:	3	203	+68

Landing Craft Launch Distance

1. Base case: 2 NM
2. Alternate #1: 6 NM
3. Alternate #2: 8 NM
4. Alternate #3: 10 NM

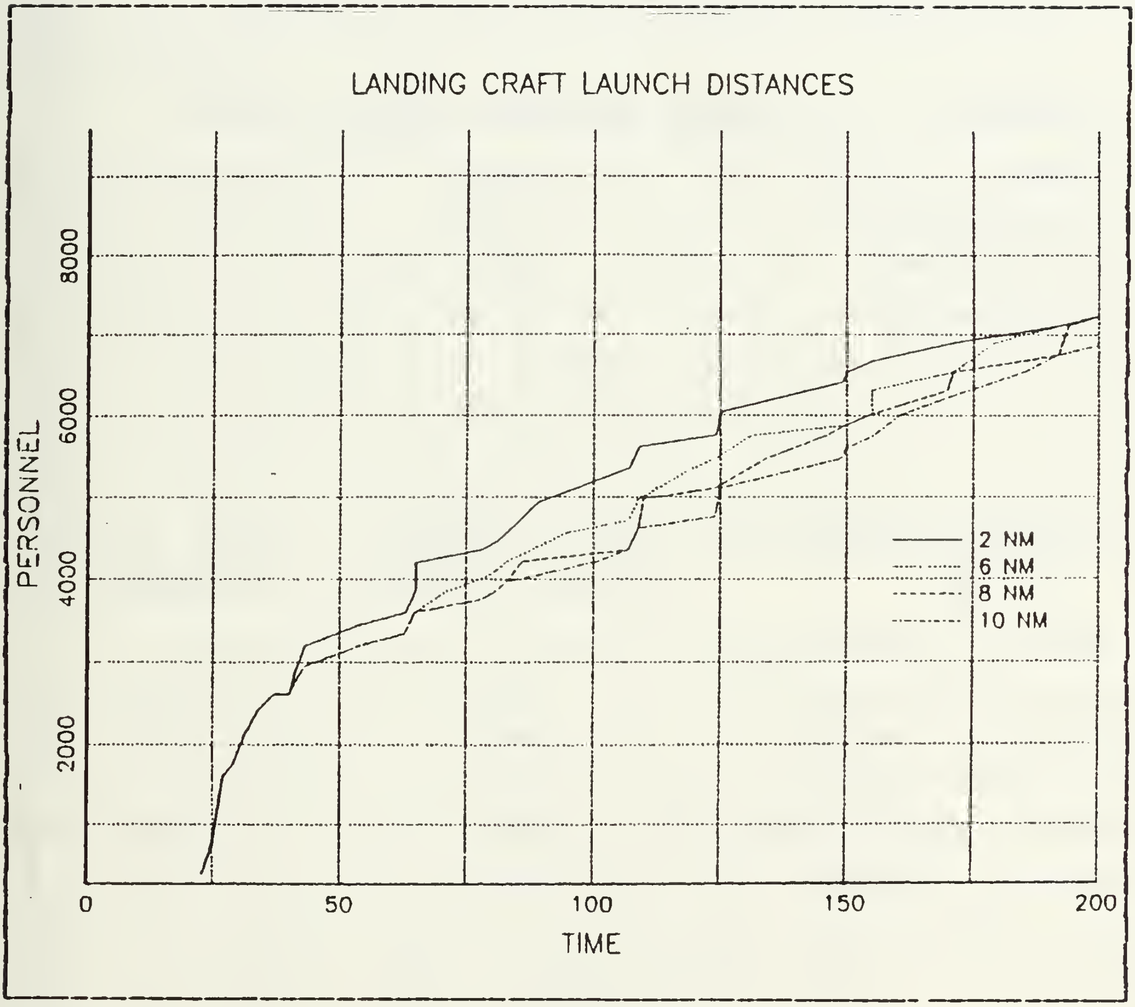


Figure 5.24 Landing Craft Launch Distamce.

TABLE XV

Sensitivity Index: Landing Craft Launch Distance

Parameter	Number personnel ashore (3 hrs.)
Base:	6894
Alternate #1:	6894
Alternate #2:	6534
Alternate #3:	6244

Change in Parameter (in NM)	Change in MOE (# troops ashore)	Sens. Index (MOE/para.)
Base - Alt #1: 4	0	0
Alt #1 - Alt #2: 2	360	180
Alt #2 - Alt #3: 2	290	145
Alt #1 - Alt #3: 6	650	109
Base - Alt #2: 6	360	60
Base - Alt #3: 8	650	82

With the above type analysis, one could readily observe how the model reacts under different parameters. The sensitivity index is a value which provides a weighting scheme in order to rate the effects of going from one set of parameters to another. The sign of the sensitivity index indicates how the model responds. A positive sign indicates improvement in the performance of the landing plan when going from one set of parameters to another. A negative sign would indicate a decrease in the MOE, thus the plan is not improving.

The sensitivity analysis was conducted utilizing a set of base conditions from which a parameter was changed to observe the effect. Table XVI lists the parameters altered which had the greatest effect within each different set of parameters. For instance, within the set of altered parameters for beach size, the greatest effect occurred when the size of the beach was increased from 2 LVTs to 4 LVTs (465 personnel ashore / minute load time). Thus, the effect of making a change decreased as the size of the beach increased beyond a 4-LVT sized beach. The other parameters listed follow similar logic.

The parameters in Table XVI are listed in order of increasing sensitivity index. This indicates the relative impact of changing a parameter by one unit as compared to the other altered parameters. Thus, the least sensitive parameter is the change of operational deck spots from 114 to 60 (11 personnel ashore / operational deck spot). The greatest sensitive parameter is found by increasing the number of 4-helo sized landing zones from 2 to 3 (890 personnel ashore / 4-helo sized LZ).

TABLE XVI

Altered Parameters Yielding Greatest Effect in MOE

Change in Parameter	Absolute Value of Sensitivity Index (PA=Personnel ashore)
Op deck spots: 114 to 60	11 PA / deck spot
Helo launch dis.: 10 to 20 NM	37 PA / NM
Number CH-46s: 72 to 54	38 PA / CH-46
# of 6-helo sized LZs: 2 to 3	138 PA / LZ
Time factors: 6 to 10 min.	146 PA / min.
Boat launch dis: 6 to 8 NM	180 PA / NM
Number beaches: 2 to 3	203 PA / beach
LZ size: 6 to 4 CH-46s	333 PA / CH-46
Beach size: 2 to 4 LVTs	465 PA / LVT
# of 4-helo sized LZs: 2 to 3	890 PA / LZ

VI. MODEL DOCUMENTATION

A key to usability of any computer software is documentation. [Ref. 10]. The general community of computer model developers and model researchers has not been overly concerned with issues of model documentation. This has severly handicapped the utility and acceptence of many valid models, and also, allowed some invalid models to be used. However, a subgroup of computer model developers, simulation modelers, have recognized that good documentation aids in establishing the validity of simulation models and in increasing usage rates [Ref. 11: p.21]

Since a model's developers are frequently not the models's ultimate user, concise documentaion is essential for effective model use. An article by Louis Fried recommends that model documentation should at a minimum provide the user and analyst views [Ref. 12]. Proper documentaion involves the use of four types of manuals which provide information for four different classes of audiences. These consist of managers (decision makers), users, analysts, and programmers. Each manual should include some common topics such as a table of contents which can be expanded upon or modified according to requirements of the specific model. Terms of each manual should be directed towards the audience of interest. The following will briefly summarize the thrust of the four types of manuals.

In the decision-maker's environment, the main goal of a manual is to assist managers to make decisions. To accomplish this end, the manual must describe the model and its applications to managers (those who sponsored the model and those who may be interested in a developed model). The manual should provide managers with sufficient information to permit them to accurately assess the model input requirements (time, money, and other resources), available outputs, and the accuracy and precision of the results. Decision makers can use this manual to justify the employment of the model and in evaluation of the results. [Ref. 13: p.2]

A user is interested mainly in deriving results from a model for specific applications. The user's manual must contain sufficient information to allow the users to employ the model intelligently and accurately. The user, although occasionally unfamiliar with computer techniques, must be informed of the model's logical structure, general simulation approach, and any assumptions and limitaions affecting the model's applicability. It is not necessary to have details of the analysis or programming beyond the preparation of input data and interpretaion of model results. [Ref. 13: p.2]

The primary purpose of a programmer's manual is for maintenance and modification of the model. Since a programmer must correct any errors discovered during model usage and possibly convert the model for use on another computer, the guidance provided by the manual should be on two levels. The micro level should contain details of what happens in each routine thus providing facts about what occurs in each of the parts of the model on an individual basis. The macro level gives an overall discussion of the processing techniques used and their relationships. Thus, it provides a grasp of how everything is tied together highlighting areas vulnerable to change. [Ref. 11: p.3]

The analysts manual is the source document for an analyst who is primarily interested in the analytical techniques and algorithms used in a model. This manual should explain the equations used in the model and the methods used for verification and validation. It is not necessary to include details such as input and output formats or programming details involving language syntax. [Ref. 13: p.3]

The decision on which types of manuals are required depends upon the individual model and should be made on a case-by-case basis. Although the main audience for the ship-to-shore model will be decision-makers and users, there exists the need for the programmer's manual as well. The analyst's manual would not offer anything different from the other three, therefore it was deemed unnecessary. The documentation for this model will be of a combined format with sections directed at the respective audiences.

Each section will be based on recommended outlines found in [Ref. 13] with whatever modifications are necessary. The sections will be self-contained and pertain to each of the three classes of audience to facilitate ease of use. The entire model documentation is found in appendices A through F and can be removed as a separate unit.

VII. CONCLUSIONS

A. SUMMARY

The proper construction of a simulation model is a formidible task. It involves many facets other than the writing of code which is supposed to reflect a real life system. There are numerous activities required prior to the programming stage which are vastly more important. Definition of the problem is probably the most critical step in the process and should involve a great deal of effort. All too often a piece of software is developed in a vacuum without regard to the user and the actual problem which is to be solved. This tends to reduce the crediblility of simulations and their usage.

This thesis attempted to demonstrate the proper procedure in the construction of a model. Although, the model was already in existence, the procedure outlined in chapter II was employed as much as possible. In some respects it was more difficult to start with a nearly finished product, learn the code, dissect the logic, flow chart the model, and then work forward through the model building steps. However, the primary purpose of this research was to complete the amphibious ship-to-shore model for use on a microcomputer. To accomplish that end it was necessary to start with the existing simulation.

Based on the material presented in this thesis, it is felt by the author that SHIPSHOR is basically a good representation of the ship-to-shore movement of an amphibious landing. From the results of the validation and sensitivity analysis and in conjunction with the author's experience, it is concluded that SHIPSHOR has a great deal of potential and is a viable decision aid. It also has room for expansion and continued improvements. The next section will address the areas for future enhancements and applicability.

B. FUTURE ENHANCEMENTS

As discussed in the beginning of this thesis, model development is a continuous process. During the conduct of research and analysis of SHIPSHOR, many areas for possible future enhancements were realized. Some of the more prominant ones are:

- More sensitivity analysis
- Improve any area discovered during additional sensitivity analysis
- Elimination of some of the limitations of the model that were encountered
- Employ stochastic modelling of the various time factors which are currently expected values
- Use the Basic Compiler to improve the speed of execution of the model
- Add a stochastic attrition subroutine option
- Automate the wave and serial assignment table which forms the basis for the landing plan file

C. FUTURE APPLICATIONS

The main reason for the development of SHIPSHOR was for use in a study which was to assist procurement planning. There was nothing specific planned beyond that objective. Currently, most U.S. models are designed for aiding analysis involved in hardware procurement and force size determination. However, there is a broad range of uses for simulations which have been overlooked by U.S. military organizations.

The Soviets have for many years utilized models as part of an integrated system of decision aids for the commander. SHIPSHOR is a type of model which could lend insight into operational planning for the commander and his staff. Once the model has gained the necessary confidence of the potential users, it could be used to help a commander make decisions regarding various options he may be considering. Choices such as the number of LZs, size of LZs, number of beaches, number of helicopters, etc. could be run through the model to observe the effects on the MOE of choice. Then, based on those results, the commander can make decisions for the landing.

Another possible application is related to predictive planning. The model could be used as a gauge to determine if the landing is proceeding as desired. If the actual landing is not progressing according to the predictions of the model, the commander could make adjustments to get back on his schedule. Of course, any use of this model will be depend on its reputation and credibility. A good reputation will only be formed after continued validation and modification.

APPENDIX A

GLOSSARY OF ACRONYMS

Acronym	Definition
CH-46:	A twin-turbine powered, tandem rotor assault transport helicopter. It has a cruising speed of 120 kts. Under ideal conditions it has a max payload of 4280 lbs. The max number of troops (the primary payload) it carries is 17.
CH-53D:	A twin-turbine powered, single main rotor assault cargo helicopter. It has a cruising speed of 150 kts. Under ideal conditions it has a max payload of 13761 lbs. The max number of troops it carries is 57. The normal payload of the CH-53 is cargo.
CH-53E:	Same characteristics as the CH-53D with the exception of the payload. The max payload of this aircraft is 16000 lbs.
LARC:	A wheeled amphibious vehicle. It can carry approximately ten troops. The primary mission of this vehicle is for use on the beach by the unit that controls the beach organization.
LCM-6:	Landing craft, mechanized. This is a flat bottomed boat used to transport vehicles and troops to shore. The 6 designates the size of the craft.
LCM-8:	Same type of landing craft as the LCM-6 with the exception of the size. The 8 indicates that it is a larger craft than the LCM-6. It can carry a tank where the LCM-6 cannot.

LCU: Landing craft, utility. Carried on larger amphibious ships although it is a self contained unit. This is the largest of the flat-bottomed landing craft used to discharge troops and vehicles ashore. It can carry two tanks.

LHA: Helicopter assault ship. It has a helicopter flight deck as well as a loading dock for landing craft. The ship actually floods its well deck in order to launch the landing craft. It displaces about 30000 tons and can carry 2000 combat troops. It is approximately the size of a WWII aircraft carrier.

LPD: Landing ship with internal loading docks for launching landing craft. It has a small two spot flight deck for helicopter operations, however does not carry its own aircraft. It displaces 17000 tons and can carry 900 troops.

LPH: Helicopter assault ship without loading docks. It has extensive medical facilities. Since it has no loading docks, it can not carry heavy equipment. It displaces 18000 tons and can carry 2000 troops. The first of the class was actually a converted WWII aircraft carrier.

LSD: Landing ship with internal loading docks for landing craft. It has a small one spot flight deck for resupply. It displaces about 13000 tons and can carry 350 troops.

LST: Landing ship, tank. Originally designed in WWII, it can land tanks and other vehicles directly onto the beach through a bow ramp. It has a small one spot flight deck for resupply. LSTs displace 8200 tons and can carry about 350 troops.

IVT: Landing vehicle, tracked. This is truly an amphibious landing craft that has tracks for movement off the beach. The current model, LVTP-7, is designed for carrying 25 troops from the amphibious ships into land combat. It is completely enclosed to provide all-around armor protection from shrapnel and small arms.[7]

---

[7] The above definintions are found in [Ref. 1] and [Ref. 7].

APPENDIX B

DECISION-MAKER'S MANUAL

A. TABLE OF CONTENTS

B. Introduction 92

C. Model Description 93

 1. Capabilities 93

 2. Input/output Classes 95

 3. Assumptions and Limitations 97

D. Development and Experimentaion 99

 1. Development History 99

 2. Verification and Validation 99

E. Current and Additional Applications 100

 1. Current Use 100

 2. Additional Applications 101

F. References 101

B. INTRODUCTION

The ship-to-shore model was initiated by the Analysis Support Branch of the Marine Corps Development Center, Quantico, Va. in support of the Amphibious Lift Study. The original model was delivered in an unusable form which prompted revision by the author. The purpose of the model is to simulate the movement of units of various sizes from amphibious shipping to shore via surface and helicopter transportation.

The purpose of the decision-maker's manual is to communicate to managers the capabilities of the ship-to-shore model. To attain this, the contents of this section will describe the model, its development and experimentation, and current applications.

C. MODEL DESCRIPTION

1. Capabilities

The model, called SHIPSHOR, can be used to determine build-up of troops and firepower ashore versus time. It can also be used to compare the effects of changes in the number and size of landing zones (beaches), numbers of carriers, numbers of amphibious ships, distances from launch, deck spot utilization, and attrition. These capabilities of the model are concerned with utilization in the planning of operations as well as use during the execution of the plan. The model accomplishes the above tasks through a series of subroutines which calculate the time for certain events to occur, then executes those events sequentially. (See Figure E.1) A detailed explanation of the subroutines can be found in the programmer's section.

The ship-to-shore model can be utilized in studies involving amphibious operations to determine optimum carrier number, launch distance mixtures, deck spot utilization, etc. The model could also be employed in a planning mode to predict approximate troop build-up ashore given the unit, distance of launch, number of carriers, number of deck spots, etc. This could be useful in planning an operation as well as gauging the progress of the landing.

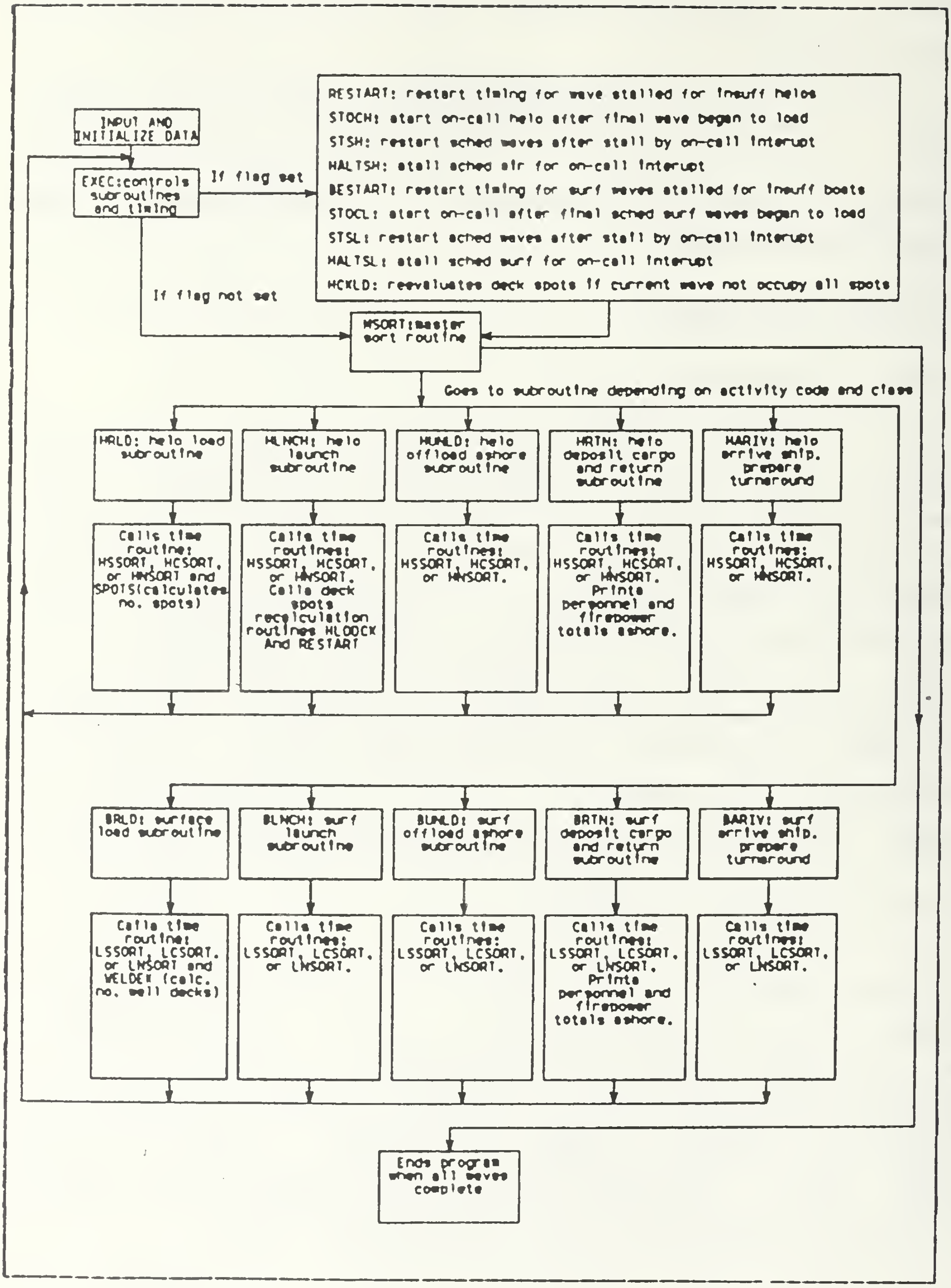

INPUT AND INITIALIZE DATA
EXEC:controls subroutines and timing
If flag set
If flag not set
RESTART: restart timing for wave stalled for insuff helos
STOCH: start on-call helo after final wave began to load
STSH: restart sched waves after stall by on-call interupt
HALTSH: stall sched air for on-call interupt
BESTART: restart timing for surf waves stalled for insuff boats
STOCL: start on-call after final sched surf waves began to load
STSL: restart sched waves after stall by on-call interupt
HALTSL: stall sched surf for on-call interupt
HCKLD: reevaluates deck spots if current wave not occupy all spots
MSORT:master sort routine
Goes to subroutine depending on activity code and class
HRLD: helo load subroutine
HLNCH: helo launch subroutine
HUNLD: helo offload ashore subroutine
HRTN: helo deposit cargo and return subroutine
HARIV: helo arrive ship, prepare turnaround
Calls time routine: HSSORT, HCSORT, or HNSORT and SPOTS(calculates no. spots)
Calls time routines: HSSORT, HCSORT, or HNSORT. Calls deck spots recalculation routines HLOOCK And RESTART
Calls time routines: HSSORT, HCSORT, or HNSORT.
Calls time routines: HSSORT, HCSORT, or HNSORT. Prints personnel and firepower totals ashore.
Calls time routines: HSSORT, HCSORT, or HNSORT.
BRLD: surface load subroutine
BLNCH: surf launch subroutine
BUNLD: surf offload ashore subroutine
BRTN: surf deposit cargo and return subroutine
BARIV: surf arrive ship, prepare turnaround
Calls time routine: LSSORT, LCSORT, or LNSORT and WELDEK (calc. no. well decks)
Calls time routines: LSSORT, LCSORT, or LNSORT.
Calls time routines: LSSORT, LCSORT, or LNSORT.
Calls time routines: LSSORT, LCSORT, or LNSORT. Prints personnel and firepower totals ashore.
Calls time routines: LSSORT, LCSORT, or LNSORT.
Ends program when all waves complete

Figure B.1 Flow Chart of Model Routines.

2. Input/Output Classes

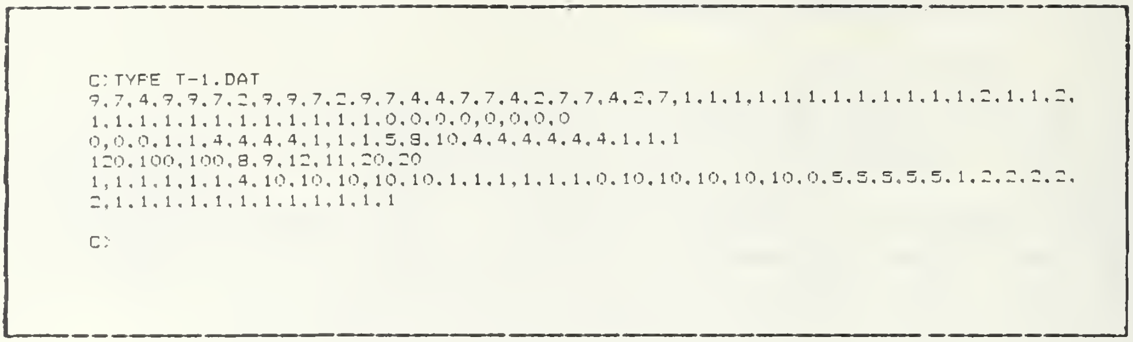
Input data is broken down into three categories. The largest group of data is found in the landing plan file. The file requires the manual construction of serials and assault waves in a specific format. The format is found in the 'help' file (Appendix E) and the user's manual (Appendix B). An example of the file is listed in Figure B.2 .

```
1S,317,128,33,35,1878,A,A,8,I,R-1,615
1S,348,128,33,35,1878,A,A,8,I,R-2,615
1S,378,128,33,35,1878,A,A,8,I,R-3,615
2S,317,128,33,35,1878,A,A,8,I,R-1,615
2S,348,128,33,35,1878,A,A,8,I,R-2,615
2S,378,128,33,35,1878,A,A,8,I,R-3,615
3S,317,280,66,70,3560,A,B,9,I,R-1,1230
3S,348,280,66,70,3560,A,B,9,I,R-2,1230
3S,383,280,66,70,3560,A,B,9,I,R-3,1230
4S,1021,50,948,5476,37414,A,B,8,I,R-1,360
4S,1025,50,948,5476,37414,A,B,8,I,R-2,360
4S,1030,50,948,5476,37414,A,B,8,I,R-3,360
5S,329,112,42,34,841,A,A,7,I,R-1,651
5S,359,112,42,34,841,A,A,7,I,R-2,651
5S;389,112,42,34,841,A,A,7,I,R-3,651
6S,329,112,42,34,841,A,A,7,I,R-1,651
6S,359,112,42,34,841,A,A,7,I,R-2,651
6S,389,112,42,34,841,A,A,7,I,R-3,651
7S,1021,62,561,3276,23015,A,A,2,I,R-1,225
```

Figure B.2 Example of Landing Plan.

Two other data sources are needed to run the model. One file contains parameter values which are not subject to change and is called T-1.DAT. See 'help' file for the format. Figure B.3 is a listing of that file. A helicopter time factors table which provides expected times for various evolutions is an example of the type of data found in this file.

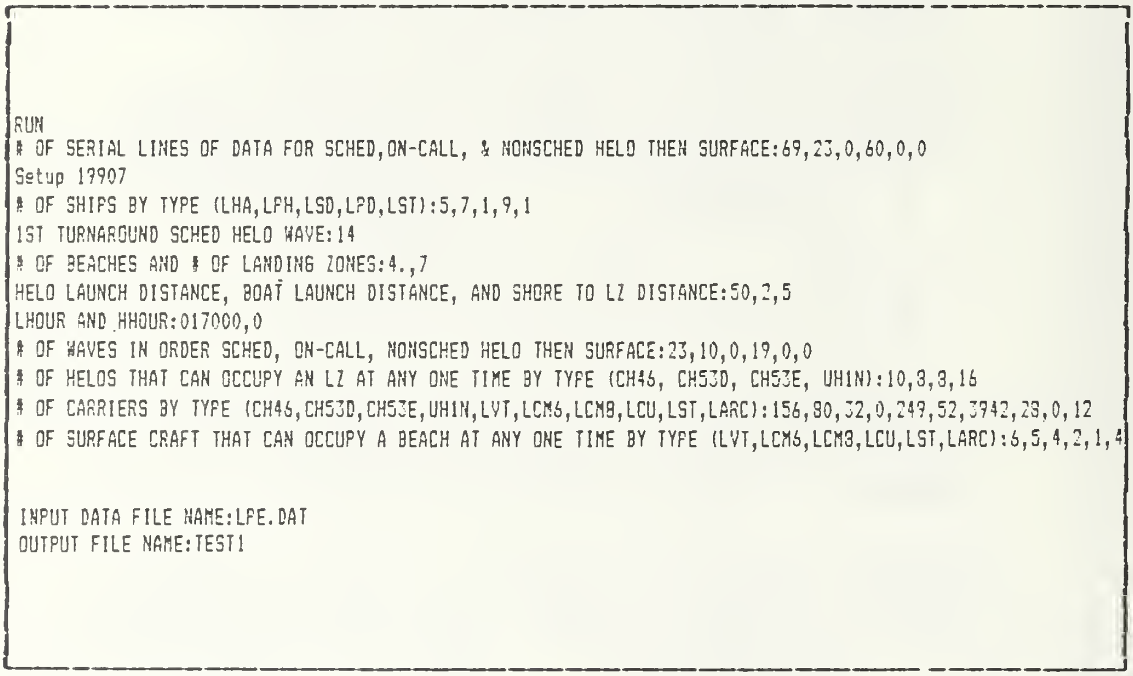
The last data source is found within the program itself as either a line of code or as an interactive query. This data is of the type that may be changed easily for various types of analysis. See 'help' file and Figure B.4 .



```
C:TYPE T-1.DAT
9,7,4,9,9,7,2,9,9,7,2,9,7,4,4,7,7,4,2,7,7,4,2,7,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,2,
1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0
0,0,0,1,1,4,4,4,4,1,1,1,5,8,10,4,4,4,4,4,4,1,1,1
120,100,100,8,9,12,11,20,20
1,1,1,1,1,1,4,10,10,10,10,10,1,1,1,1,1,1,0,10,10,10,10,10,0,5,5,5,5,5,1,2,2,2,2,
2,1,1,1,1,1,1,1,1,1,1,1,1

C>
```

Figure B.3 Example of Data File T-1.DAT.



```
RUN
# OF SERIAL LINES OF DATA FOR SCHED,ON-CALL, & NONSCHED HELO THEN SURFACE:69,23,0,60,0,0
Setup 19907
# OF SHIPS BY TYPE (LHA,LPH,LSD,LPD,LST):5,7,1,9,1
1ST TURNAROUND SCHED HELO WAVE:14
# OF BEACHES AND # OF LANDING ZONES:4.,7
HELO LAUNCH DISTANCE, BOAT LAUNCH DISTANCE, AND SHORE TO LZ DISTANCE:50,2,5
LHOUR AND HHOUR:017000,0
# OF WAVES IN ORDER SCHED, ON-CALL, NONSCHED HELO THEN SURFACE:23,10,0,19,0,0
# OF HELOS THAT CAN OCCUPY AN LZ AT ANY ONE TIME BY TYPE (CH46, CH53D, CH53E, UH1N):10,3,3,16
# OF CARRIERS BY TYPE (CH46,CH53D,CH53E,UH1N,LVT,LCM6,LCM8,LCU,LST,LARC):156,80,32,0,249,52,3942,28,0,12
# OF SURFACE CRAFT THAT CAN OCCUPY A BEACH AT ANY ONE TIME BY TYPE (LVT,LCM6,LCM8,LCU,LST,LARC):6,5,4,2,1,4


 INPUT DATA FILE NAME:LPE.DAT
 OUTPUT FILE NAME:TEST1
```

Figure B.4 Example of Interactive/Line Input Data.

The output of the model can be modified to reflect the on-going utilization. For exmaple, if the MOE is build-up ashore, the output could be personnel and firepower ashore at given times. This data could be printed or sent to a file for use in a graphics program to plot the build-up. See Figure B.5 .

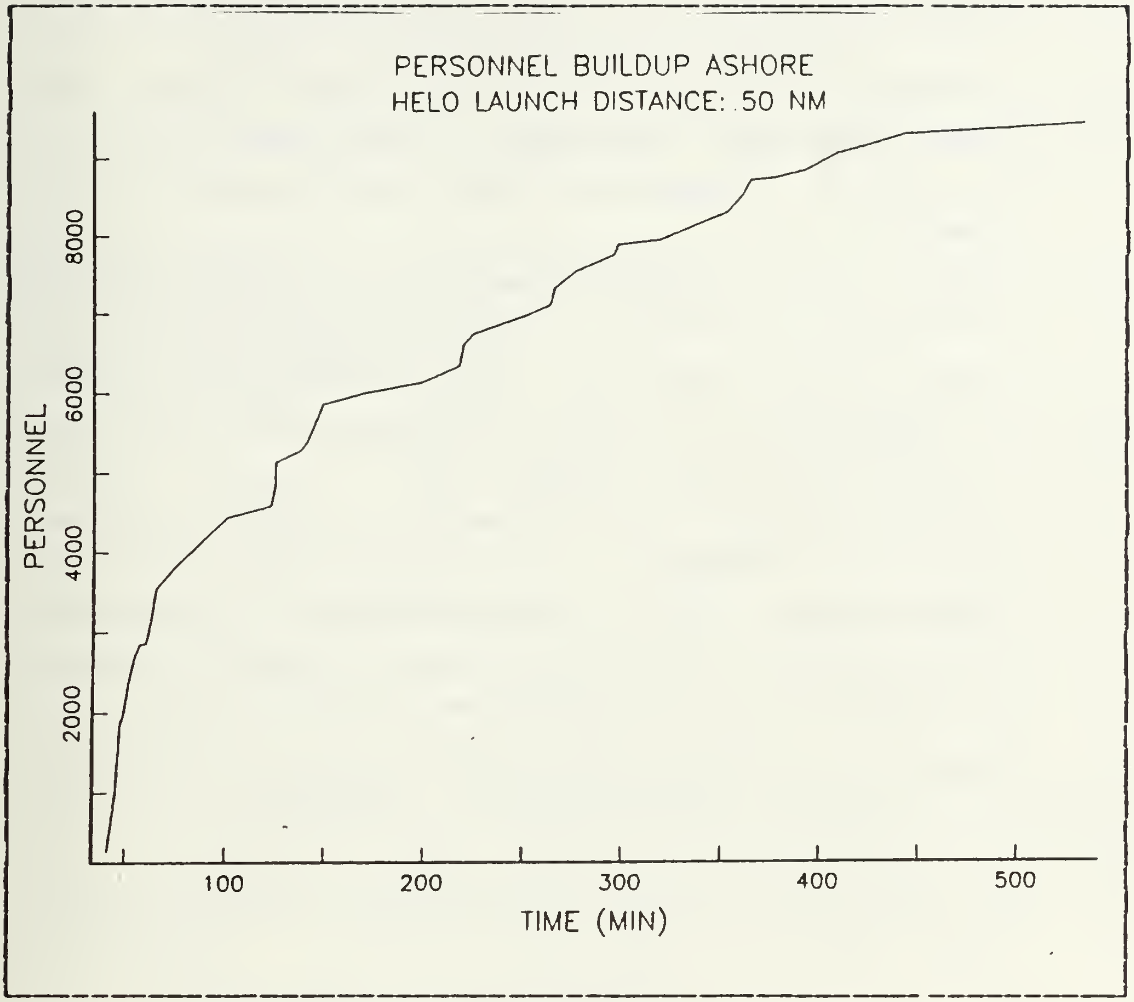


Figure B.5 Example of a Graph of Build-up Ashore.

Total time to land a particular sized force could be obtained for several different sets of input data. Then a graph could be made comparing the various sets of inputs to obtain sensitivity analysis for the desired parameters. For details of the coding requirements see the user's manual.

3. Assumptions and Limitations

Assumptions which affect the results of the model are as follows:

- Helicopter and boat time factors are expected values as opposed to stochastic values. This assumption is considered reasonable since the factors are used many times. This lends itself to the use of averages according to the central limit theorem.
- There is an attrition rate which is predetermined and set as a constant. This designates mechanical and/or combat attrition. The value used is based on experience and not on any factual data therefore it is considered an assumption.
- All time calculations are assumed to be rounded to the nearest minute.
- Environmental effects are assumed to negligible which implies load limitations of the helicopters and landing craft are based on an average day.

Limitations of the model depend highly on the model purpose. The following are the major limitations:

- In order to run the surface portion of the model, there must be a sufficient number of carriers to land the scheduled and on-call waves.
- Each helicopter wave must use the same type of helicopter. Since the model computes all activities and times based on a particular wave, the speed of the aircraft must be the same.
- Once the loads are assigned to a wave and helicopter type, they can not be switched during the simulation. Therefore, unlike the flexibility in reality of switching loads to available craft, the model does not have that capability.
- Waves and serials are not designated to a particular ship. Therefore, the simulation looks at the total number of deck/well spots for the entire

amphibious force instead of those individual ships to which craft are assigned to load.

- The current edition is limited to a maximum of four landing zones or beaches. This can be modified by making a change in the coding.

D. DEVELOPMENT AND EXPERIMENTATION

1. Development History

The ship-to-shore model is currently the first revised edition. The original was developed by General Research Company for use in the Amphibious Lift Factors Study at the Development Center Quantico, Va. The developers turned over the product in an unusable form with a paucity of documentation. The current form was revised by the author. It involved variable identification and then construction of a detailed flow chart to understand the logic of the model. Enhancements included interactive input, output to a file, a detailed documentation section, a 'help' file, and conversion to IBM disk operating system for an IBM PC.

2. Verification and Validation

The validation conducted on the model consisted of two phases. The initial phase involved the testing of a simplified landing plan which could be checked by hand calculations. The results showed that the model performed the basic computations as it was designed.

The second phase of the validation utilized the results of another, large scale simulation. Since historical data did not exist, it was necessary to compare SHIPSHOR against another simulation (CAAM) to verify the output. CAAM was used in a study conducted by the Marine Corps in 1983 (see [Ref. 9]). The method for comparison was a graphical technique in which the output from each model was used. The parameters for the simualation were similar so that a direct comparison could be made. A ten percent confidence interval was placed around the output from CAAM to provide a feel for the similarities.

The results of the above procedure produced outputs which were very close. There is no way to place a numerical value on the comparisons that would make any sense. Thus, the method employed for phase two of the validation is more subjective than objective. However, the confidence in SHIPSHOR, based on the validation conducted, should be as high as that of CAAM.

Finally, sensitivity analysis was conducted on SHIPSHOR using a variety of parameter sets to observe how the model behaved. The results were portrayed graphically which provided a feel for the performance of the simulation when conducted under different conditions. This analysis demonstrates which parameters affect the model greatly and which have a lesser influence on the results.

The details of validation and sensitivity analysis can be found in Chapter V of this thesis. The analysis was not, by any stretch of the imagination, all encompassing. That would require a considerable amount of study due to the large number of parameter combinations that might be tested. However, the analysis that was conducted can be repeated under many conditions of interest to give further enlightenment to the user of the model.

E. CURRENT AND ADDITIONAL APPLICATIONS

1. Current Use

Current interest in the ship-to-shore model exists in the plans and operations section of HQMC for amphibious lift planning. The model can provide a desk-top decision aid for immediate 'what if' type questions.

2. Additional Applications

The model could have application during an amphibious landing to determine the progress of the landing. This would also provide historical data for continued validation and refinement of the model. Additionally, utilization in the planning phase of an operation could provide a commander with estimates of the time involved to land his forces using a particular scheme of maneuver. He could investigate the effect of changing various input parameters within the model to aid in his decision process. For instance given his force size, he may want to determine the minimum number of helicopter landing zones which would give him an acceptable build-up ashore.

Automation of the construction of the wave and serial assignment tables would reduce the large amount of time required for input data preparation. This would result in a self-contained ship-to-shore model with realistic combat uses.

F. REFERENCES

See M.S. thesis by Major S. M. Ritacco, USMC, 1984, NPS.

APPENDIX C
USER'S MANUAL

A. TABLE OF CONTENTS

B. Introduction 102

C. Description of the Model 103

  1. Overview 103

  2. Methodology 109

  3. Assumptions and Limitations 115

D. Model Input Data 117

  1. General Description 117

  2. Detailed Description 120

  3. Data Collection and Maintenance 123

E. Model Output Data 126

  1. General Description 126

  2. Detailed Description 126

F. Sample Run 128

G. Variable Names and Locations 128

B. INTRODUCTION

The ship-to-shore model was initiated by the Analysis Support Branch of the Marine Corps Development Center, Quantico, Va. in support of the Amphibious Lift Study. The original model was delivered in an unusable form which prompted revision by the author. The purpose of the model is to simulate the movement of units of various sizes from amphibious shipping to shore via surface and helicopter transportation.

The purpose of the user's section is to provide nonprogamming users of the ship-to-shore model with the information necessary to use the model effectively. To accomplish this end the following section will describe the model, give details for the model input data, explain the model output data, and give some sample runs and examples.

C. DESCRIPTION OF THE MODEL

1. Overview

a. Model Identification

The model documented in this manual is named SHIPSHOR. It is designed to simulate the movement of men and equipment via helicopters and surface craft from amphibious shipping to shore during an amphibious landing. SHIPSHOR is written in Microsoft Basic and designed to be used on an IBM PC. However, it can be modified to run on any pc which can be converted to run MIcrosoft Basic formatted for MDOS. Currently SHIPSHOR has no relation to any other models.

b. Physical System Highlights

The major activities during a ship-to-shore movement for either helicopter-borne or surface-borne units are listed in Figure C.1

The first activity is the loading phase. This takes place aboard the amphibious shipping located in its designated sea echelon area. It consists of the loading of troops and/or equipment aboard helicopters or surface craft.

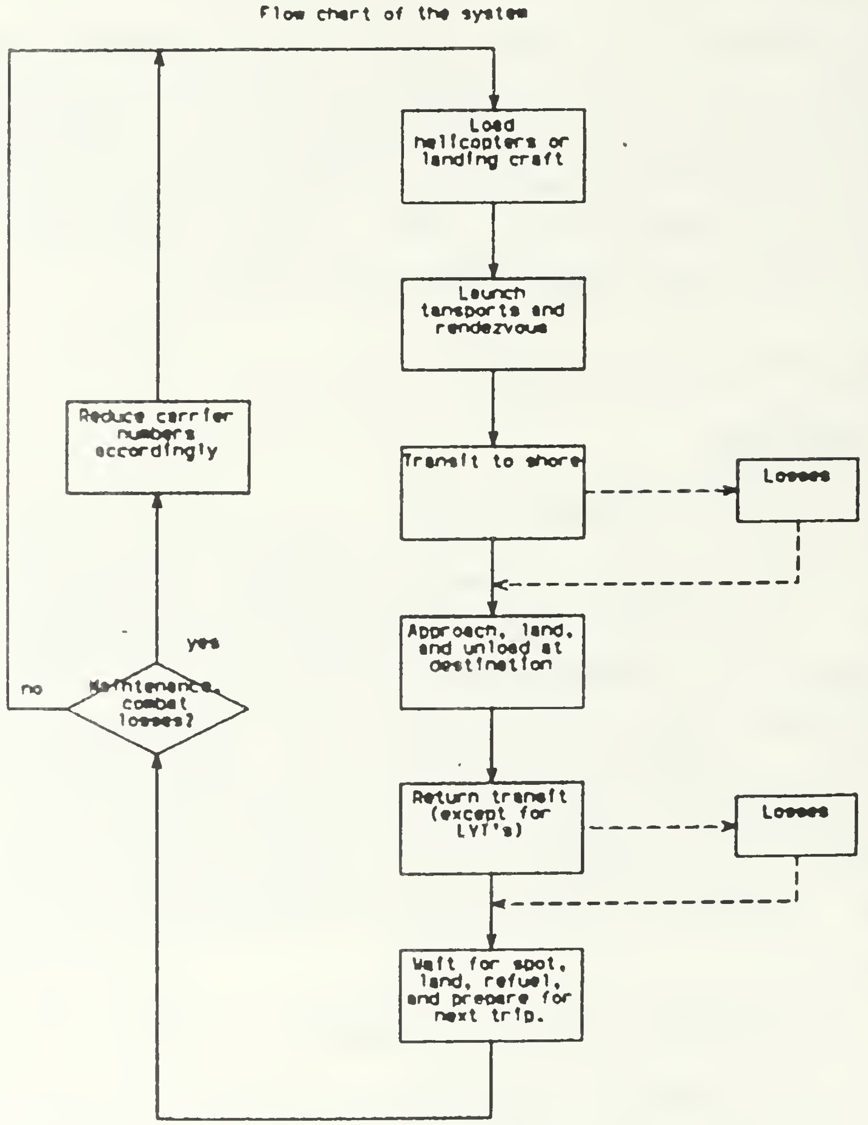

Flow chart of the system
Load helicopters or landing craft
Launch tansports and rendezvous
Transit to shore
Losses
Approach, land, and unload at destination
Return transit (except for LVT's)
Losses
Wait for spot, land, refuel, and prepare for next trip.
Reduce carrier numbers accordingly
yes
no
Maintenance, combat losses?

Figure C.1 Flow Chart of the System.

The launching phase follows loading and involves the launch and rendezvous of the carriers in order to proceed to shore as an entire wave. Once the wave is constituted it travels via prescribed routes to its destination.

Approach, landing, and off-loading transpires next at the appropriate destination. This phase of the amphibious landing is where combat attrition is most likely to occur. Both landing craft and helicopters are most vulnerable in a static situation while troops and equipment off-load.

The return transit takes place for all carriers except LVT's (amphibious tracked vehicles) which proceed inland. Upon return to amphibious shipping the transports must wait for an opening on the flight deck or the well-deck to land (dock). The vehicles will then refuel and prepare for their next trip ashore. The cycle is then repeated until all necessary assault waves are ashore.

Each class of wave (scheduled, on-call, and nonscheduled) will execute all the events listed above. The order in which each class of wave will commence the cycle is normally (1) scheduled, (2) on-call, and then (2) nonscheduled. However, on-call and nonscheduled waves can interrupt the sequence depending on the situation ashore.

c. Model Applicability

The model, called SHIPSHOR, can be used to determine build-up of troops and firepower ashore versus time. It can also be used to compare various changes in the number and size of landing zones (beaches), numbers of carriers, numbers of amphibious ships, distances from launch, deck spot utilization, and attrition. These capabilities of the model are concerned with the utilization in the planning of operations as well as use during the execution of the plan. The model accomplishes the above tasks through a series of subroutines which calculate the time for certain events to occur, then it executes those events sequentially. (See Figure C.2) A detailed explanation of the subroutines can be found in the programmer's section.

The ship-to-shore model can be utilized in studies involving amphibious operations to determine optimum carrier number, launch distance mixtures, deck spot utilization, etc. The model could also be employed in a planning mode to predict approximate troop build-up ashore given the unit, distance of launch, number of carriers, number of deck spots, etc. This could be utilized in the planning phase of an operation to provide a commander with estimates of the time involved to land his forces using a particular scheme of maneuver. He could investigate the effect of changing various input parameters within the model to aid in his decision process. For instance given his force size, he may want to determine the minimum number of helicopter landing zones which would give him an acceptable build-up ashore.

d. Input and Output

There are three basic categories of data input for SHIPSHOR. The landing plan data file is the largest and most important. It contains the wave and serial data which are such items as personnel, wave number, destination, firepower score, type of carrier, class of the wave, etc. The second data file is called T-1.dat and is made up of all the parameter data which is not likely to change often. The third source of input data is either interactive input or line input (to change parameters within code). This data consists of parameters which will be changed often during sensitivity analysis.

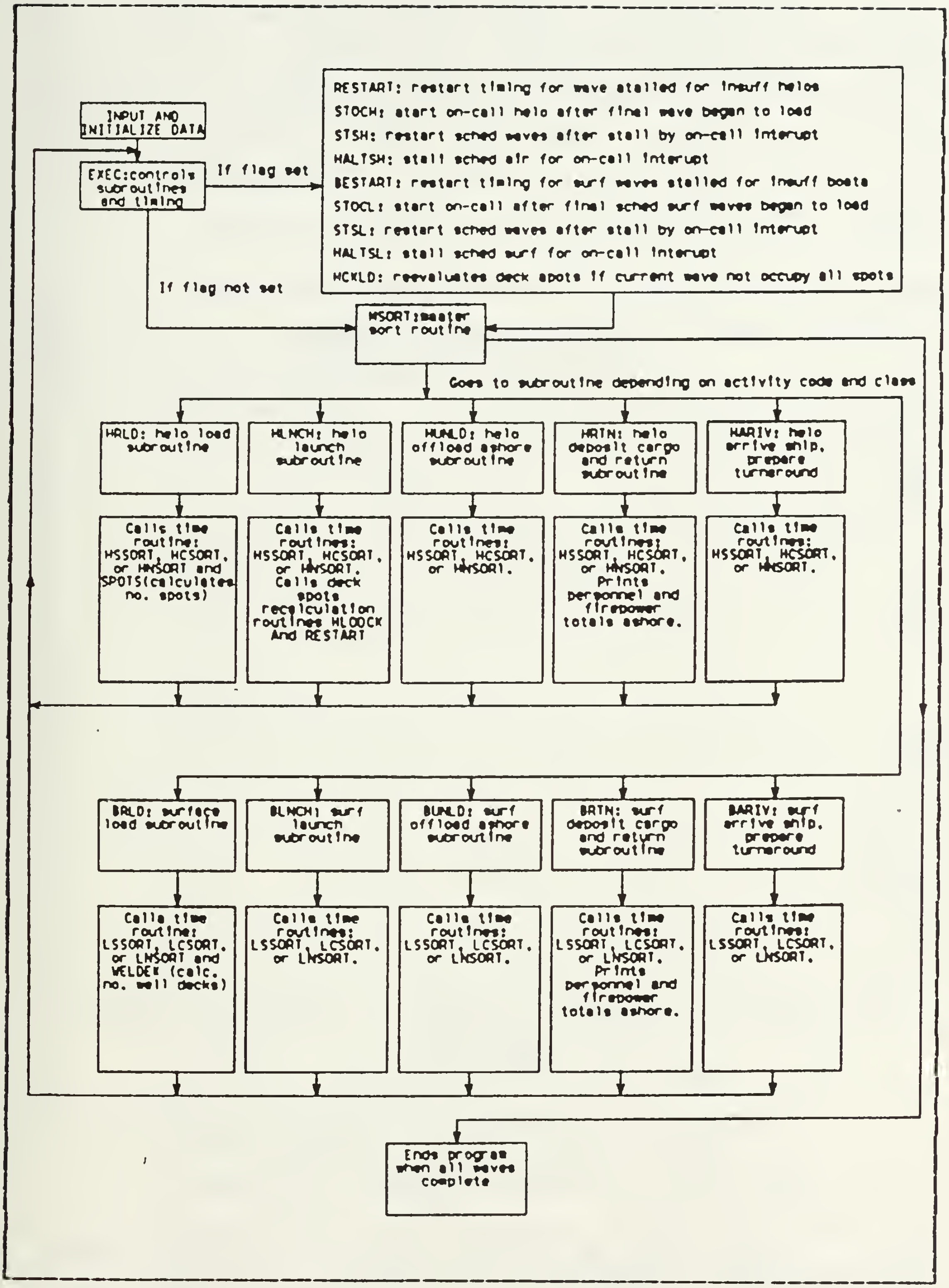

INPUT AND INITIALIZE DATA
EXEC:controls subroutines and timing
If flag set
RESTART: restart timing for wave stalled for insuff helos
STOCH: start on-call helo after final wave began to load
STSH: restart sched waves after stall by on-call interupt
HALTSH: stall sched air for on-call interupt
BESTART: restart timing for surf waves stalled for insuff boats
STOCL: start on-call after final sched surf waves began to load
STSL: restart sched waves after stall by on-call interupt
HALTSL: stall sched surf for on-call interupt
HCKLD: reevaluates deck spots if current wave not occupy all spots
If flag not set
MSORT:master sort routine
Goes to subroutine depending on activity code and class
HRLD: helo load subroutine
HLNCH: helo launch subroutine
HUNLD: helo offload ashore subroutine
HRTN: helo deposit cargo and return subroutine
HARIV: helo arrive ship, prepare turnaround
Calls time routine: HSSORT, HCSORT, or HNSORT and SPOTS(calculates no. spots)
Calls time routines: HSSORT, HCSORT, or HNSORT. Calls deck spots recalculation routines HLDOCK And RESTART
Calls time routines: HSSORT, HCSORT, or HNSORT.
Calls time routines: HSSORT, HCSORT, or HNSORT. Prints personnel and firepower totals ashore.
Calls time routines: HSSORT, HCSORT, or HNSORT.
BRLD: surface load subroutine
BLNCH: surf launch subroutine
BUNLD: surf offload ashore subroutine
BRTN: surf deposit cargo and return subroutine
BARIV: surf arrive ship, prepare turnaround
Calls time routine: LSSORT, LCSORT, or LNSORT and WELDEK (calc. no. well decks)
Calls time routines: LSSORT, LCSORT, or LNSORT.
Calls time routines: LSSORT, LCSORT, or LNSORT.
Calls time routines: LSSORT, LCSORT, or LNSORT. Prints personnel and firepower totals ashore.
Calls time routines: LSSORT, LCSORT, or LNSORT.
Ends program when all waves complete

Figure C.2 Flow Chart of the Model.

The output of the model can be modified to reflect the on-going utilization. For exmaple, if the MOE is build-up ashore, the output could be personnel and firepower ashore at given times. This data could be printed or sent to a file for use in a graphics program to plot the build-up. See Figure C.3 .

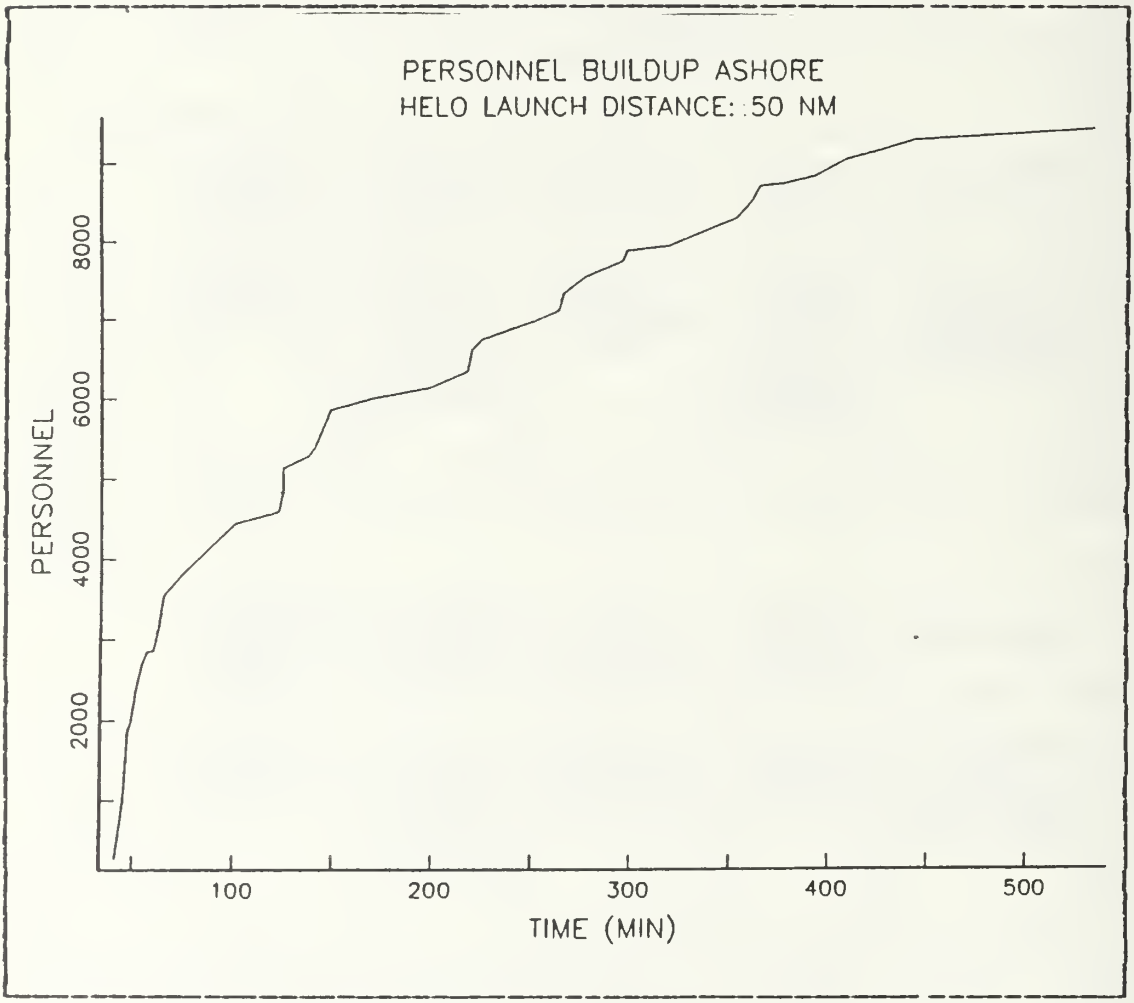


Figure C.3 Graph of Build-up Ashore.

Total time to land a particular sized force could be obtained for several different sets of input data.

Then a graph could be made comparing the various sets of inputs to obtain sensitivity analysis for the desired parameters. For details of the coding requirements, see the programmer's or user's section.

2. Methodology

a. Physical System Details

The following gives more details for the physical system being modelled and is an elaboration of Physical System Highlights section:

- Loading phase: The loading phase can only take place if there are sufficient carriers available to load the entire wave. Also, there must be the requisite number of deck (well-deck) spots to accomodate the carriers which are to load. Once those conditions have been met the first activity can be completed.
- Launch phase: No additional details.
- Unloading phase: The conduct of this phase depends upon two restrictions. The beach or landing zone must be large enough to accomodate the current wave. If the preceeding wave is still present, the landing area may not be large enough which will cause a delay. Also, the current wave may be too large by itself which means only a limited number of boats or helicopters can off-load while the other vehicles await an opening.
- Return phase: No additional details.
- Arrive back phase: This phase can be completed when there is a sufficient number of deck spots (well or flight) to accomodate the returning

vehicles. Once the wave of vehicles has landed the cycle can repeat itself.

b. Model Logic and Data Flow

The model utilizes 35 subroutines to attempt to translate reality into mathematical computations. A diagram of the model in very general terms is found in Figure C.2 The first subroutine in the program reads the data from the wave and serial file and the parameter file. Also the variable data is entered within the first section of the program. The second subroutine initializes the arrays and clocks to be used by the timing and activity subroutines, based on the information found the wave data file. These subroutines essentially complete set-up and initialization of the model.

Once the model enters the EXEC subroutine it continues the time/event procedure until all waves have been landed. The EXEC subroutine is the control routine in which all waves eventually pass. Based on the class of the wave (air or surface) and the current activity (load, launch, unload, return, or arrive back ship), the EXEC subroutine will send the wave in question to the proper subroutine for manipulation.

Prior to going to one of the activity subroutines, the program queries whether certain flags have been set. If they have been set, one of the following routines could be called. Five are for helicopter waves and four are for surface waves. They are:

- RESTART(helo) / BESTART(surface): Restart timing for waves that have been stalled by the lack of deck spots (helo or surface). Since in reality there are not enough spots available to load all of the assault carriers at one time, subsequent waves may be delayed. This routine restarts timing after such a delay.

- STOCH(helo) / STOCL(surface): These subroutines start the on-call waves to load after the final scheduled wave has begun to load.

- HALTSH(helo) / HALTSL(surf): Stall remaining scheduled waves when one or more on-call waves interrupt the scheduled category. As in reality, on-call waves can break in when called for. The model allows flags to be set to simulate this process.

- SISH(helo) / STSL(surface): Start scheduled waves that have been stalled due to on-call waves interrupting the sequence. More than one on-call wave can interrupt at different times so the HALTSH(L)/SISL(L) subroutines can be called more than once.

- HCKLD(helo): Reevaluates deck spot employment if the current wave does not occupy all spots available. It calls WSPOT which calculates the number of spots to be used in the current wave.

One last subroutine will be called prior to the activity subroutines. This is the MSORT routine which acts as the master sort routine. It takes the lowest time from six other time/event sort routines and that becomes the next event which is processed by the appropriate activity subroutine.

There are three time sort routines common to the helicopter activities and three time sort routines common to the surface activities. They all sort through the waves identifying the wave with the earliest time for the next event. That wave is placed on the top of the queue from which MSCRT then picks the lowest time out of the six time sort routines. Listed below are the sort subroutines:

- HSSORT: Scheduled helicopter wave sort routine.

- HCSORT: On-call helicopter wave sort routine.

- HNSORT: Nonscheduled helicopter wave sort routine.

- LSSORT: Scheduled surface wave sort routine.

- LCSORT: On-call surface wave sort routine.

- LNSORT: Nonscheduled surface wave sort routine.

The activity subroutines process each wave based on its class, activity code, and category (scheduled, on-call, or nonscheduled). There are five major activity subroutines for each class (air or surface). Each activity subroutine calls other secondary routines and the appropriate time sort routine as determined by the wave category. Once the wave has completed an activity subroutine, it returns to the EXEC subroutine and the cycle begins again. Thus, each wave will eventually cycle through five subroutines.

The following is a summary of the activity subroutine and their supporting routines:

- HRLD: The purpose is to load (or attempt to load) a helicopter wave from available flight deck spots of the shipping. Loading will not be initiated if (1) no deck spots are currently available or if (2) insufficient helicopters of the proper type to carry the serials are present at the time. HRLD calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT) and subroutine SPOTS. The SPOTS routine identifies whether all the deck spots are curently being employed for loading. A flag is set to indicate whether deck spots are available or not.

- HLNCH: This routine launches a helicopter wave from ships and determines the time it will arrive at the landing zone(s). It determines the number of deck spots made available to subsequent waves and restores those spots. Also it reevaluates the loading time for a wave being loaded which had employed all of the remaining deck spots. HLNCH calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT) and HLODCK and RESTART. HIODCK sets up signals that will initiate a reevaluation of the deck spot employment in the event that the current wave was not employing the last available deck spot. Those signals will trigger the use of HCKLD which is called from EXEC. If the current wave was occupying the remaining deck spots, a flag is set to call RESTART from EXEC. RESTART allows subsequent waves to employ deck spots which will be vacated by the current wave.

- HUNLD: HUNID determines whether or not landing zones are still occupied by previous waves and if so delays the landing of the current wave until the pertinent zones are free to use. Also it calculates the length of time necessary for this wave to unload (constrained by the size of the landing zone). HUNID calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT).
- HRTN: The purpose is to print a running total of the personnel ashore and the firepower ashore. Also HRTN calculates the time that the wave will return to the area of the shipping and hence be available for formation of subsequent waves. It calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT).

- HARIV: HARIV returns helicopters to the helicopter pools for use in forming subsequent waves after degrading the number of helicopters to reflect maintenance losses. It also initiates the loading process for subsequent waves. HARIV calls the appropriate time sort routine (HSSORT, HCSORT, or HNSORT).

The surface activity subroutines operate in basically the same manner (the first letter for surface activities is B instead of H). Since the following routines are so similar to the helicopter routines, only the differences will be pointed out.

- BRLD: The purpose is to load the landing craft in a similar fashion to the HRLD for helicopters. It calls the appropriate time sort routines (LSSORT, LCSORT, or LNSORT) and the subroutine WELDEK. WELDEK calculates the well decks to be used by landing craft of each wave and indicates whether there are sufficient openings or not.

- BLNCH: This routine launches the surface waves from the ships. It calls the appropriate time sort routine (LSSORT, LCSORT, or LNSORT).

- BUNLD: BUNLD determines if the beach is open for landing and offloads the wave if it is possible. It calls the appropriate time sort routine (LSSORT, LCSORT, or LNSORT).

- BRTN: BRTN prints the running total of personnel ashore and the firepower ashore. Also, it returns the boats to the ships. It calls the appropriate time sort routine (LSSORT, LCSORT, or LNSORT).

- BARIV: This routine returns the boats to the boat pools for subsequent use. It calls the appropriate time sort subroutine (LSSORT, LCSORT, or LNSORT).

Once each wave has cycled through the five helicopter activities or the five surface activities, they are assigned a very large value in their time array. This prevents them from recycling. The program will end when the last nonscheduled surface wave has gone through all activities. At that point, all personnel in the wave data file will be ashore.

3. Assumptions and Limitations

a. System Related

Assumptions which affect the results of the model are as follows:

- Helicopter and boat time factors are expected values as opposed to stochastic values. This assumption is considered reasonable since the factors are used many times. This lends itself to the use of averages according to the central limit theorem.
- There is an attrition rate which is predetermined and set as a constant. This designates mechanical and/or combat attrition. The value used is based on experience and not on any factual data therefore it is considered an assumption.
- All time calculations are assumed to be rounded to the nearest minute.
- Environmental effects are assumed to negligible which implies load limitations of the helicopters and landing craft are based on an average day.

Limitations of the model depend highly on the model purpose. The following are the major limitations:

- In order to run the surface portion of the model, there must be a sufficient number of carriers to land the scheduled and on-call waves.
- Each helicopter wave must use the same type of helicopter. Since the model computes all activities and times based on a particular wave, the speed of the aircraft must be the same.
- Once the loads are assigned to a wave and helicopter type, they can not be switched during the simulation. Therefore, unlike the flexibility in reality of switching loads to available craft, the model does not have that capability.
- Waves and serials are not designated to a particular ship. Therefore, the simulation looks at the total number of deck/well spots for the entire amphibious force instead of those individual ships to which craft are assigned to load.
- There is a max of four landing zones or beaches that can be used for any landing. This can be modified if more are desired by making a change in the code.

b. Model Parameters

The model parameters must agree with each other. For example, the parameter TA (number scheduled helo waves) must reflect the number of waves found in the serial data input file. If a parameter value is changed, caution must be exercised to ensure other values are not affected in an adverse manner. This becomes very apparent when the landing plan data file is changed. That change affects the number of wave parameters, the number of serial parameters, and a host of other values. The 'help' file gives guidance along these lines.

c. Output Limitations

As with any model, results from the simulation must be used judiciously. The model output is only as good as its validation and verification. Even then, no model is perfect and results must be used cautiously. SHIPSHOR must undergo a rigorous validation and subsequent modifications in a series of actual amphibious operations before its predictive capabilities are verified as reliable.

d. Restrictions on Model Use

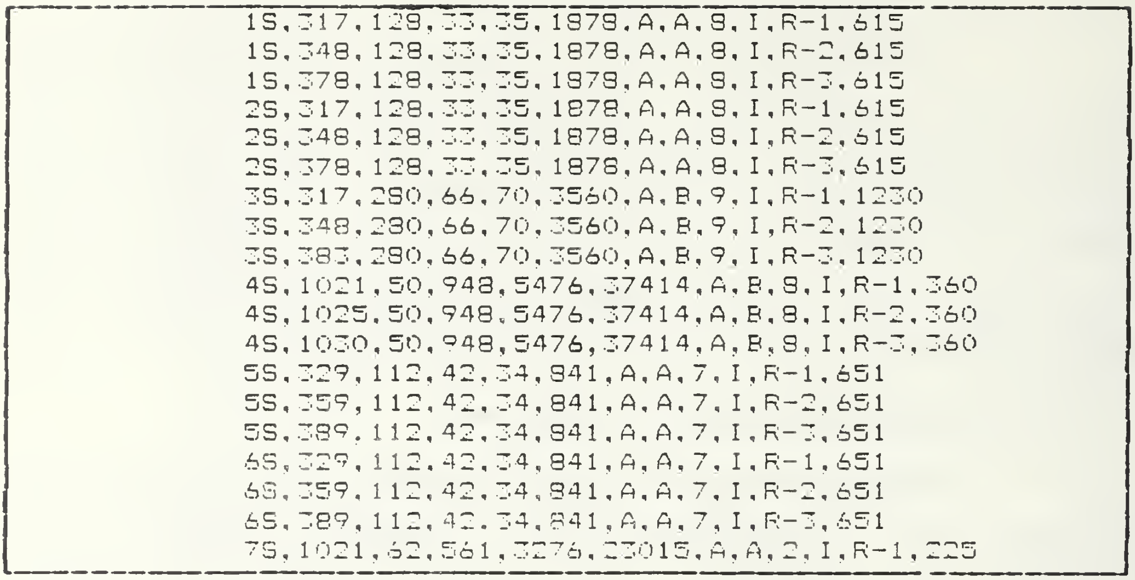
As with any model the current model should be considered as an approximation of a complex process which contains many variables. It is not designed to produce results on which policy can be based. It is, however, to be used to give an estimate where only a bad guess would exist otherwise.

D. MODEL INPUT DATA

1. General Description

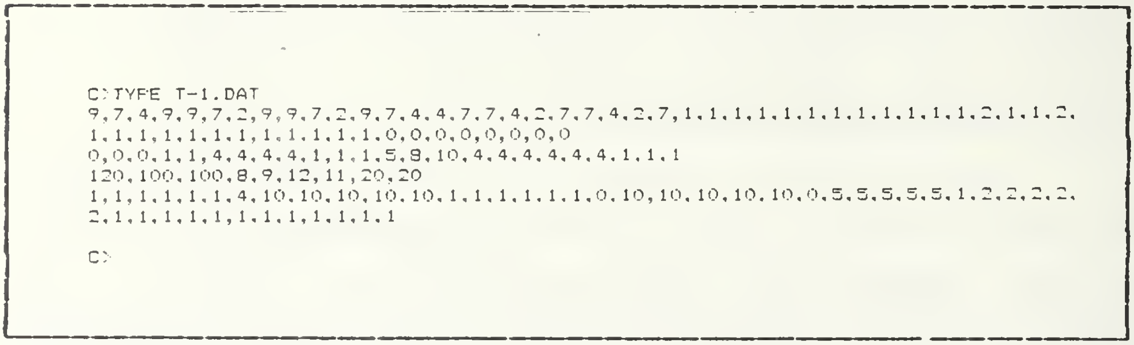
Input data is broken down into three categories. The largest group of data is found in the landing plan file. The file requires the manual construction of serials and assault waves in a specific format. The format is found in the 'help' file and the user's manual. An example of this file is listed in Figure C.4 .

Two other data sources are needed to run the model. One file contains parameter values which are not subject to change and is called T-1.dat. See 'help' file for the format. Figure C.5 is a listing of the file. A helicopter time factors table which provides expected times for various evolutions is an example of the type of data found in this file.



```
1S,317,128,33,35,1878,A,A,8,I,R-1,615
1S,348,128,33,35,1878,A,A,8,I,R-2,615
1S,378,128,33,35,1878,A,A,8,I,R-3,615
2S,317,128,33,35,1878,A,A,8,I,R-1,615
2S,348,128,33,35,1878,A,A,8,I,R-2,615
2S,378,128,33,35,1878,A,A,8,I,R-3,615
3S,317,280,66,70,3560,A,B,9,I,R-1,1230
3S,348,280,66,70,3560,A,B,9,I,R-2,1230
3S,383,280,66,70,3560,A,B,9,I,R-3,1230
4S,1021,50,948,5476,37414,A,B,8,I,R-1,360
4S,1025,50,948,5476,37414,A,B,8,I,R-2,360
4S,1030,50,948,5476,37414,A,B,8,I,R-3,360
5S,329,112,42,34,841,A,A,7,I,R-1,651
5S,359,112,42,34,841,A,A,7,I,R-2,651
5S,389,112,42,34,841,A,A,7,I,R-3,651
6S,329,112,42,34,841,A,A,7,I,R-1,651
6S,359,112,42,34,841,A,A,7,I,R-2,651
6S,389,112,42,34,841,A,A,7,I,R-3,651
7S,1021,62,561,3276,23015,A,A,2,I,R-1,225
```

Figure C.4 Example of Landing Plan Data File.

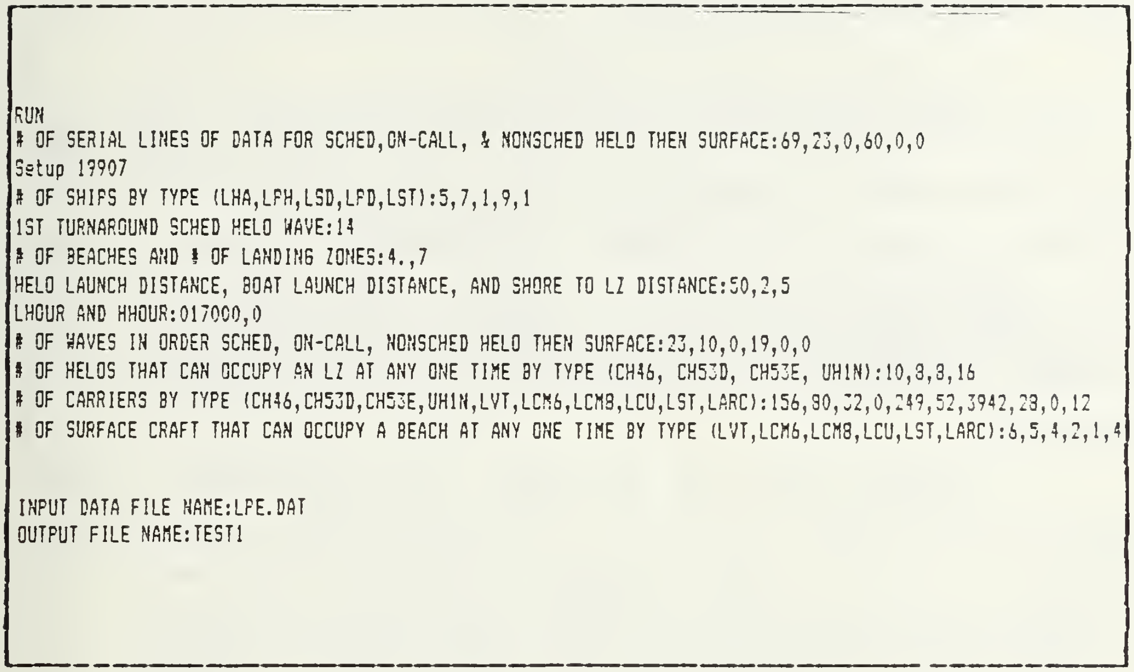


```
C>TYPE T-1.DAT
9,7,4,9,9,7,2,9,9,7,2,9,7,4,4,7,7,4,2,7,7,4,2,7,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,2,
1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0
0,0,0,1,1,4,4,4,4,1,1,1,5,8,10,4,4,4,4,4,4,1,1,1
120,100,100,8,9,12,11,20,20
1,1,1,1,1,1,4,10,10,10,10,10,1,1,1,1,1,1,0,10,10,10,10,10,0,5,5,5,5,5,1,2,2,2,2,
2,1,1,1,1,1,1,1,1,1,1,1,1

C>
```

Figure C.5 Example of Data File T-1.DAT.

The last data source is found within the program itself as either a line of code or as an interactive query. This data is of the type that may be changed easily for various types of analysis. See 'help' file and Figure C.6 . The data sources are summarized in Table XVII .



```
RUN
# OF SERIAL LINES OF DATA FOR SCHED,ON-CALL, & NONSCHED HELO THEN SURFACE:69,23,0,60,0,0
Setup 19907
# OF SHIPS BY TYPE (LHA,LPH,LSD,LPD,LST):5,7,1,9,1
1ST TURNAROUND SCHED HELO WAVE:14
# OF BEACHES AND # OF LANDING ZONES:4.,7
HELO LAUNCH DISTANCE, BOAT LAUNCH DISTANCE, AND SHORE TO LZ DISTANCE:50,2,5
LHOUR AND HHOUR:017000,0
# OF WAVES IN ORDER SCHED, ON-CALL, NONSCHED HELO THEN SURFACE:23,10,0,19,0,0
# OF HELOS THAT CAN OCCUPY AN LZ AT ANY ONE TIME BY TYPE (CH46, CH53D, CH53E, UH1N):10,8,8,16
# OF CARRIERS BY TYPE (CH46,CH53D,CH53E,UH1N,LVT,LCM6,LCM8,LCU,LST,LARC):156,80,32,0,249,52,3942,28,0,12
# OF SURFACE CRAFT THAT CAN OCCUPY A BEACH AT ANY ONE TIME BY TYPE (LVT,LCM6,LCM8,LCU,LST,LARC):6,5,4,2,1,4


 INPUT DATA FILE NAME:LPE.DAT
 OUTPUT FILE NAME:TEST1
```

Figure C.6 Example of Interactive/Line Input Data.

TABLE XVII

Summary of Input Data Sources

Category	Name	Data Media
Landing plan	LPE.DAT	Disk File
Fixed parameters	T-1.DAT	Disk File
Interactive	---	Terminal
Program lines	---	Listing

2. Detailed Descriptions

a. Landing Plan Data File

The landing plan file contains the wave and serial data which is processed by the model. It has type of carriers, number of personnel, wave number, destination, etc. for each serial going ashore. Its purpose is to provide the model with the troop unit, equipment, and transportation information which will be 'moved ashore' by the simulation.

Each line in the data file contains twelve pieces of information which the model requires. These are explained as part of an example listed in figure C.7 . Figure C.7 is a part of a 'help' file which located on the same disk as the main program and is for use in constructing the data file. A copy of the 'help' file is located in Appendix E.

b. Fixed Parameter Data File T-1.DAT

This data file contains parameter values which are not subject to constant change. The three major tables of this type are the helicopter time factors, boat time factors, and operational deck spot tables. The first two provide the expected times for various evolutions to utilize and the last gives the deck spots under various conditions. Tables XVIII, XIX, and XX list the data which are read from T-1.DAT. The purpose of this file is to read data into various arrays within the program which will be used to compute event times.

The first two lines of the T-1.DAT file refer to Table XX The data is listed in the file row by row from the table. Thus, the first twelve values come from row one of the table. The next line contains data from table XVIII It is listed in the same manner as Table XX except the first

```
 type land the

     THE FOLLOWING IS AN EXAMPLE OF THE DATA INPUT FOR ALL WAVE
INFORMATION.  THE FIRST LINE OF LETTERS AND NUMBERS IS AN ACTUAL LINE OF
DATA FOUND IN THE LPE.DAT FILE. THE CORRESPONDING LETTERS WILL EXPLAIN
EACH COLUMN OF DATA.
     ...........................................................
                  1S,317,128,33,35,1878,A,A,8,I,R-1,615
                  :   :   :   :  :   :   : : : :  :   :
                  A , B , C ,D ,E , F   ,G,H,I,J, K , L
     ...........................................................

A:      THE NUMBER INDICATES THE WAVE # AND THE LETTER REFERS TO THE
        CLASS OF THE WAVE (S=SCHEDULED, O=ON-CALL, N=NON-SCHED).MORE
        THAN ONE LINE OF DATA (CALLED A SERIAL) CAN BE OF THE SAME WAVE
        NUMBER.  EACH SERIAL RUNS SEQUENTIALLY FROM 1 TO THE LAST SERIAL
        THE CLASS.

B:      RESERVED FOR FUTURE USE.

C:      NUMBER OF PERSONNEL IN PARTICULAR SERIAL OF A PARTICULAR WAVE

D-F:    RESERVED FOR FUTURE USE.

G:      MODE OF TRANSPORTATION. (A=HELO, S=SURFACE)

H:      CATEGORIES OF CARRIERS. A THROUGH K CORRESPOND TO 1-11. (1=CH46E
        2=CH53D, 3=CH53E, 4=UHIN, 5=LVT, 6=LCM-6, 7=LCM-8, 8=LCU, 9=LST,
        10=LARC, 11=LCAC). CARRIER TYPE MUST BE THE SAME FOR THE SAME
        WAVE.

I:      NUMBER OF CARRIERS IN THE SERIAL

J:      TYPE LOAD (I=INTERNAL, E=EXTERNAL, C=COMBINATION)

K:      DESTINATION: (B-1 THROUGH  B-4 REFERS TO BEACHES #1 THROUGH #4
        AND R-1 THROUGH R-4 REFERS TO LANDING ZONES #1 THROUGH #4)

L:      WEAPONS UNIT VALUE (WUV-FIREPOWER SCORE)
END OF FILE

R; T=0.01/0.09 16:55:59

.
```

Figure C.7 Format for the Landing Plan File.

four values come from the first row of the table. Line four lists three helicopter speeds then six landing craft speeds. The vehicles for which these speeds are listed are: CH-53, CH-46, UH1N, LVT, LCM-6, LCM-8, LCU, LST, and LARC. The last two lines of the file contain data from Table XIX They are also listed by row. (see Figure C.5)

TABLE XVIII

Helicopter Time factors in Minutes

Helo wave operation	Internal	External	Combined
	Type of Load		
(1) Land in zone	0	0	0
(2) Disembark troops (cargo)	1	1	4
(3) Takeoff/rendezvous from LZ	4	4	4
(4) Maneuver/land ship	1	1	1
(5) Refuel/embark troops (cargo)	5	8	10
(6) Takeoff/rendezvous from ship	4	4	4
(7) Maneuver/land in LZ	4	4	4
(8) Safety clearance time	1	1	1

c. Interactive Data

Although this source is not a file, it is the final means used to assign values to the remaining variables. Once the first two files are prepared, the model is loaded into the computer and initiated. The program will query the user to input data for the parameters which are most likely to change during any type of analysis. An alternative is to change the code within the program itself where some of these values are found. The parameters that are requested by the interactive input are listed in the Table XXI .

TABLE XIX
Boat Time Factors in Minutes

	Boat type						
Boat wave operation	LVT	LCM-6	LCM-8	LCU	LST	LARC	LCAC
(1) Land at beach (or LZ)	1	1	1	1	1	1	1
(2) Disembark troops/load	4	10	10	10	10	10	15
(3) Rendezvous for return	1	1	1	1	1	1	0
(4) Dock time to ship (well deck)	0	10	10	10	10	10	10
(5) Refuel/embark troops/load	0	5	5	5	5	5	15
(6) Circle and rendezvous	1	2	2	2	2	2	1
(7) Land at beach	1	1	1	1	1	1	1
(8) Safety clearance time	1	1	1	1	1	1	1

3. Data Collection and Maintenance

a. Data Sources

Two documents must be completed prior to construction of the landing plan files. These documents are the Helicopter Wave and Serial Assignment Table and the Landing Craft and Amphibious Assignment Table ([Ref. 10: p.29]). They are normally prepared during the planning phase of the amphibious operation. Once these documents are completed, the necessary data exists for compilation of the landing plan file.

The preparation of the preceeding two documents is a very large, time consuming endeavor. It is extremely complex and its intricacy goes up exponentially as the size of the unit. Unfortunately, an automated system for this task does not exist at this time.

TABLE XX

Operational Deck Spots

Ship type	Internal Loading A	B	C	D	External Loading A	B	C	D	Combination Loading A	B	C	D
LHA(#1)	9	7	4	9	9	7	2	9	9	7	2	9
LPH(#2)	7	4	4	7	7	4	2	7	7	4	2	7
LSD(#3)	1	1	1	1	1	1	1	1	1	1	1	1
LPD(#4)	2	1	1	2	1	1	1	1	1	1	1	1
LST(#5)	1	1	1	1	0	0	0	0	0	0	0	0

Helo types: A=CH46E B=CH53D C=CH53E D=UH1N

Sources for the remaining data file and interactive parameters depend on the make-up of the fleet, number of helicopters and landing craft available, distance to launch, etc. Fixed values such as the helicopter time factors are expected values found in Marine Corps Publications such as reference 7 dealing with amphibious operations.

b. Updating Procedures

Each time a new landing plan file is used, all the parameters pertaining to the number of waves, the number of serials, the number of landing craft for scheduled and on-call waves, the name of the landing plan file, etc. must be changed to conform to the new data. If this is not accomplished an error will occur identifying the line in the code where an inconsistency was found. Most of those values

TABLE XXI

Interactive Parameters

Order of Request	Parameter Name	Definition
1	TA	Sched helo serial #
2	CA	On-call helo serial #
3	NA	Nonsched helo serial #
4	TS	Sched surf serial #
5	CS	On-call surf serial #
6	MS	Nonsched surf serial #
7-12	NUS(I)	Ship type (five types)
13	TRNH1	1st turnaround wave
14	NNB	Number of beaches
15	NLZ	Number of LZs
16	ASADIS	Helo launch distance
17	LODDIS	LVT launch distance
18	LZDIS	Shore to LZ distance
19	HSWAVE	Sched wave #
20	HCWAVE	On-call wave #
21	HNWAVE	Nonsched wave #
22	LSWAVE	Sched wave #
23	LCWAVE	On-call wave #
24	LNWAVE	Nonsched wave #
25-28	HGF(I)	# helos max in LZ by type (4 types)
29-38	AT(I)	# carriers by type (10 types)
39-44	LGF(I)	# boats max at beach by type (6 types)

are found in the interactive mode which helps to prevent those types of errors.

Other parameters can be changed for sensitivity analysis purposes. Since there is such a large number of parameters, all possible combinations have not been experimented with. Therefore, there exists the chance that a certain combination of parameters will produce an error.

E. MODEL OUTPUT DATA

1. General Description

The output of the model can assume a variety of forms depending upon the immediate purpose of the run. A detailed print out of each wave and its status at various times can be obtained by changing the PRINT command to IPRINT in line 3692 of the program as explained in Appendix C. Currently this output is suppressed to expedite the time to run the simulation.

Another form of output is the output data file. The current edition of the model sends the time, total personnel ashore, and total firepower ashore to a file for use in graphic presentations. The program will prompt the user to input that data file name during the intialization of the model.

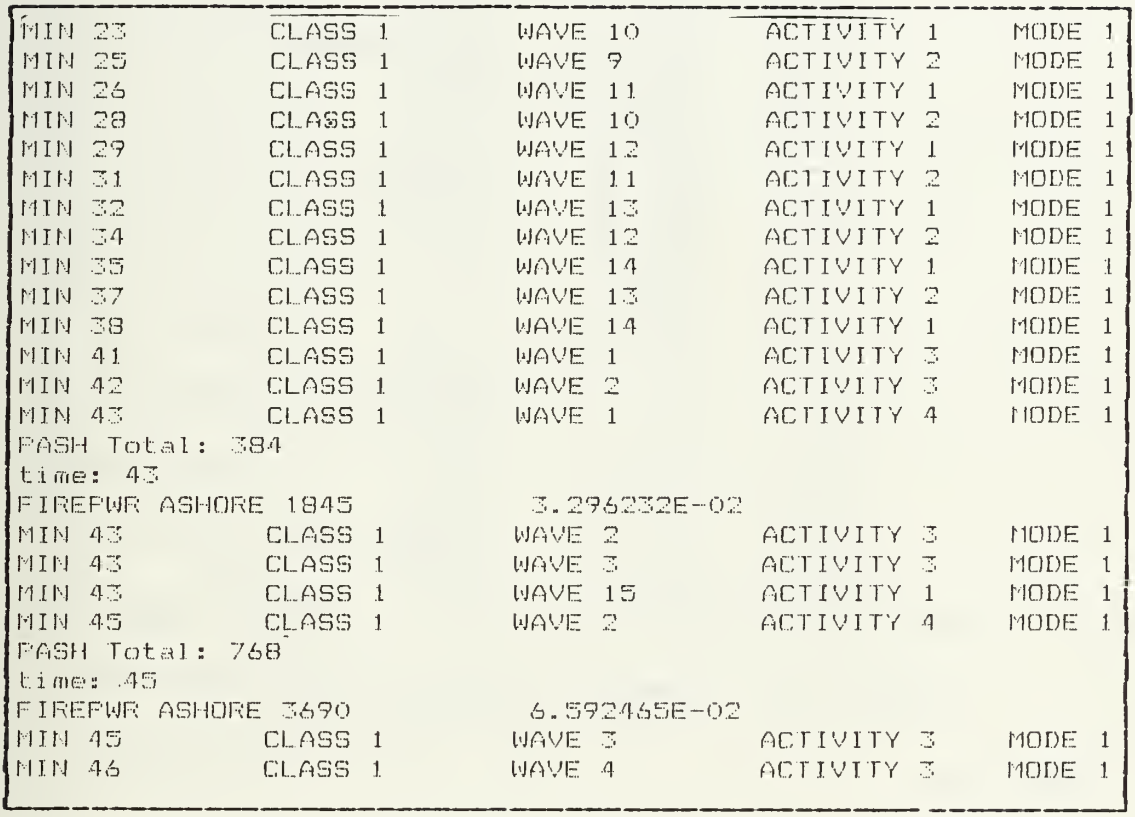
2. Detailed Description

a. Printed Output Data

The following terms used in Table C.8 are defined:

- Wave: The wave number currently being processed
- Activity: 1=load, 2=launch, 3=unload, 4=return, 5=arrive. Describes the landing activity currently being conducted.
- Mode: 1=helo, 2=surface. Describes whether the current wave is helo or surface.

- Min: The time of the current activity.
- PASH: Personnel ashore.
- FIRPWR: Firepower ashore and percent firepower ashore.



```
MIN 23          CLASS 1         WAVE 10         ACTIVITY 1      MODE 1
MIN 25          CLASS 1         WAVE 9          ACTIVITY 2      MODE 1
MIN 26          CLASS 1         WAVE 11         ACTIVITY 1      MODE 1
MIN 28          CLASS 1         WAVE 10         ACTIVITY 2      MODE 1
MIN 29          CLASS 1         WAVE 12         ACTIVITY 1      MODE 1
MIN 31          CLASS 1         WAVE 11         ACTIVITY 2      MODE 1
MIN 32          CLASS 1         WAVE 13         ACTIVITY 1      MODE 1
MIN 34          CLASS 1         WAVE 12         ACTIVITY 2      MODE 1
MIN 35          CLASS 1         WAVE 14         ACTIVITY 1      MODE 1
MIN 37          CLASS 1         WAVE 13         ACTIVITY 2      MODE 1
MIN 38          CLASS 1         WAVE 14         ACTIVITY 1      MODE 1
MIN 41          CLASS 1         WAVE 1          ACTIVITY 3      MODE 1
MIN 42          CLASS 1         WAVE 2          ACTIVITY 3      MODE 1
MIN 43          CLASS 1         WAVE 1          ACTIVITY 4      MODE 1
PASH Total: 384
time: 43
FIREPWR ASHORE 1845              3.296232E-02
MIN 43          CLASS 1         WAVE 2          ACTIVITY 3      MODE 1
MIN 43          CLASS 1         WAVE 3          ACTIVITY 3      MODE 1
MIN 43          CLASS 1         WAVE 15         ACTIVITY 1      MODE 1
MIN 45          CLASS 1         WAVE 2          ACTIVITY 4      MODE 1
PASH Total: 768
time: 45
FIREPWR ASHORE 3690              6.592465E-02
MIN 45          CLASS 1         WAVE 3          ACTIVITY 3      MODE 1
MIN 46          CLASS 1         WAVE 4          ACTIVITY 3      MODE 1
```

Figure C.8   Example of Output.

b. Output Data File

The name of the output file is determined by the user when the prompt queries for output data file name. The data which is sent to the output file consists of three columns. The first column is the time in minutes of the off-loading ashore of a particular wave. The second and third columns contain the total personnel ashore and the total firepower ashore for the time listed in the first column. An example is found in figure C.9

```
  29.00   384.00      0.03
  37.00   768.00      0.07
  41.00  1016.00      0.10
  47.00  1856.00      0.16
  55.00  2006.00      0.18
  63.00  2342.00      0.22
  65.00  2694.00      0.27
  71.00  3030.00      0.30
  73.00  3216.00      0.31
  77.00  3216.00      0.31
  85.00  3552.00      0.32
  87.00  3693.00      0.32
  89.00  4053.00      0.37
  93.00  4309.00      0.39
  99.00  4565.00      0.41
 101.00  4845.00      0.46
 111.00  4957.00      0.46
 125.00  5247.00      0.50
 139.00  5402.00      0.50
 149.00  5762.00      0.54
 163.00  5866.00      0.54
 185.00  6128.00      0.54
 215.00  6266.00      0.55
 218.00  6469.00      0.57
 225.00  6607.00      0.57
```

Figure C.9 Example of Output File.

F. SAMPLE RUN

See Chapter V of thesis.

G. VARIABLE NAMES AND LOCATIONS

See Section F of Appendix E.

APPENDIX D

PROGRAMMER'S MANUAL

A. TABLE OF CONTENTS

B. Introduction 130

C. Model specifications 131

D. Model description 132

1. Overview 132

2. General logic flow 132

E. Description of routines 134

F Data base description 158

G. Source Listing 163

H. Glossary of variables 164

I. Model test results 164

B. INTRODUCTION

The ship-to-shore model was initiated by the Analysis Support Branch of the Marine Corps Development Center, Quantico, Va. in support of the Amphibious Lift Study. The original model was delivered in an unusable form which prompted revision by the author. The purpose of the model is to simulate the movement of units of various sizes from amphibious shipping to shore via surface and helicopter transportation.

The purpose of this manual is to provide programmer personnel of SHIPSHOR with the information necessary to effectively maintain and modify the model. Although the flow diagram of the entire model exists, due to its bulk it was not included in this thesis. However, the author will maintain a copy of the flow diagram if one has need of it.

C. MODEL SPECIFICATION

The following provides a short summary of the model specifications.

1. Purpose

The purpose of SHIPSHOR is to simulate an amphibious ship-to-shore movement.

2. Host System

IBM Personnel computer

3. Processing requirements

IBM PC with 64K RAM and a printer

4. Language

Microsoft Basic

5. Capabilities

- Observe build-up of personnel and firepower ashore
- Sensitivity analysis for operational planning
- Time sequence of each landing activity for each wave
- Predict troop build-up to gauge progress of landing

6. Limitations

- Must have sufficient surface carriers to land scheduled and on-call waves

- Each helo wave must use the smae type of helicopter
- Loads assigned to a particular wave cannot be switched

D. MODEL DESCRIPTION

1. Overview

The model documented in this manual is named SHIPSHOR. It is designed to simulate the movement of men and equipment via helicopters and surface craft from amphibious shipping to shore during an amphibious landing. SHIPSHOR is written in Microsoft Basic and designed to be used on an IBM PC. However, it can be modified to run on any pc which can be converted to run MIcrosoft Basic formatted for MDOS. Currently SHIPSHOR has no relation to any other models.

2. General logic flow

The model utilizes 35 subroutines to attempt to translate reality into mathematical computations. A diagram of the model in very general terms is found in Figure D.1 The first subroutine in the program reads the data from the wave and serial file and the parameter file. Also the variable data is entered within the first section of the program. The second subroutine initializes the arrays and clocks to be used by the timing and activity subroutines, based on the information found the wave data file. These subroutines essentially complete set-up and initialization of the model.

Once the model enters the EXEC subroutine it continues the time/event procedure until all waves have been landed. The EXEC subroutine is the control routine in which all waves eventually pass. Based on the class of the wave (air or surface) and the current activity (load, launch, unload, return, or arrive back ship), the EXEC subroutine

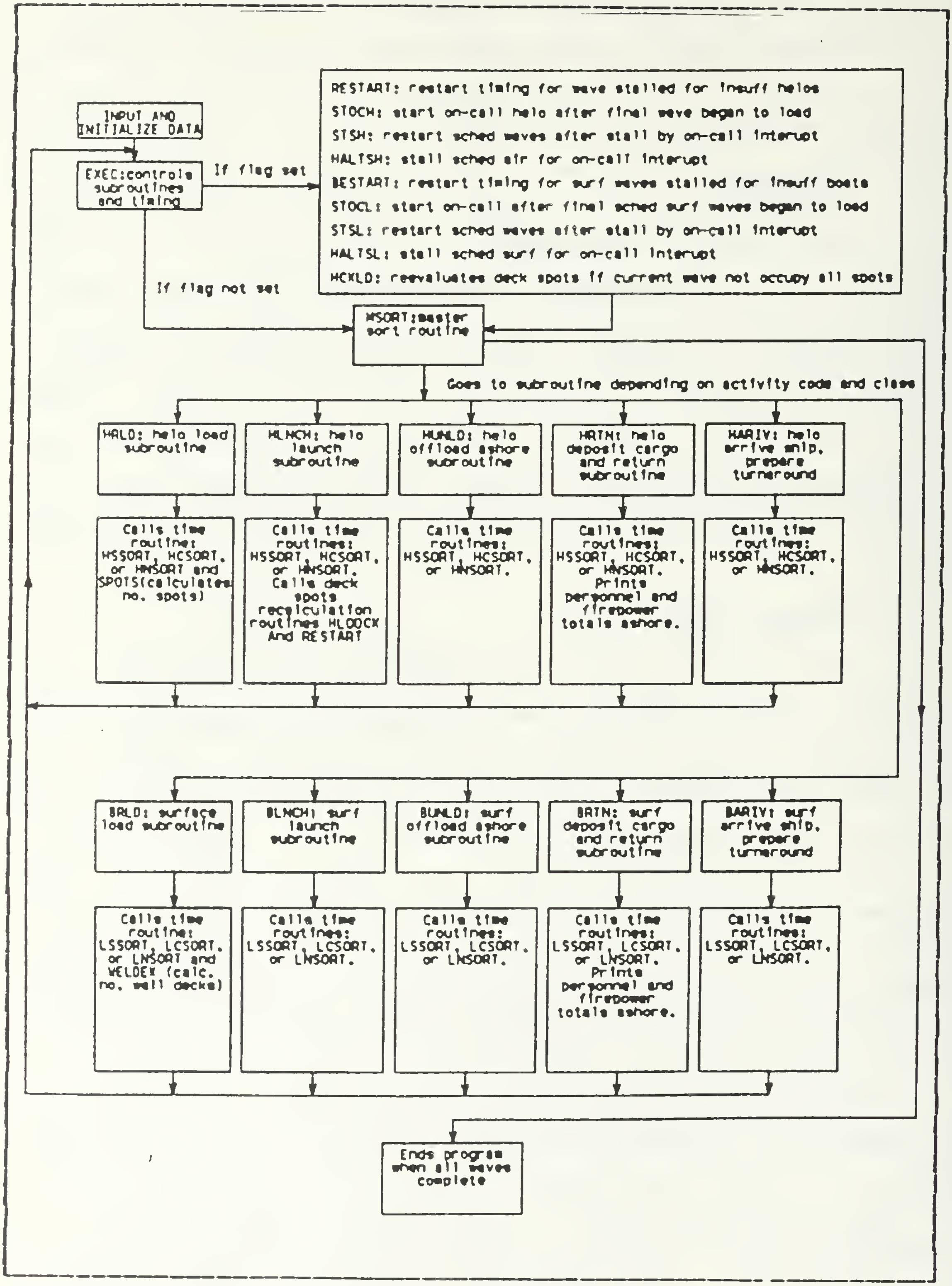

INPUT AND INITIALIZE DATA
EXEC:controls subroutines and timing
If flag set
RESTART: restart timing for wave stalled for insuff helos
STOCH: start on-call helo after final wave began to load
STSH: restart sched waves after stall by on-call interupt
HALTSH: stall sched air for on-call interupt
BESTART: restart timing for surf waves stalled for insuff boats
STOCL: start on-call after final sched surf waves began to load
STSL: restart sched waves after stall by on-call interupt
HALTSL: stall sched surf for on-call interupt
HCXLD: reevaluates deck spots if current wave not occupy all spots
If flag not set
MSORT:master sort routine
Goes to subroutine depending on activity code and class
HALD: helo load subroutine
HLNCH: helo launch subroutine
HUNLD: helo offload ashore subroutine
HRTN: helo deposit cargo and return subroutine
HARIV: helo arrive ship, prepare turnaround
Calls time routine: HSSORT, HCSORT, or HNSORT and SPOTS(calculates no. spots)
Calls time routines: HSSORT, HCSORT, or HNSORT. Calls deck spots recalculation routines HLDOCK And RESTART
Calls time routines: HSSORT, HCSORT, or HNSORT.
Calls time routines: HSSORT, HCSORT, or HNSORT. Prints personnel and firepower totals ashore.
Calls time routines: HSSORT, HCSORT, or HNSORT.
BRLD: surface load subroutine
BLNCH: surf launch subroutine
BUNLD: surf offload ashore subroutine
BRTN: surf deposit cargo and return subroutine
BARIV: surf arrive ship, prepare turnaround
Calls time routine: LSSORT, LCSORT, or LNSORT and WELDEK (calc. no. well decks)
Calls time routines: LSSORT, LCSORT, or LNSORT.
Calls time routines: LSSORT, LCSORT, or LNSORT.
Calls time routines: LSSORT, LCSORT, or LNSORT. Prints personnel and firepower totals ashore.
Calls time routines: LSSORT, LCSORT, or LNSORT.
Ends program when all waves complete

Figure D.1 Flow Chart of the Model.

will send the wave in question to the proper subroutine for manipulation.

Prior to going to one of the activity subroutines, the program queries whether certain flags have been set. If they have been set, one of nine routines could be called. Five are for helicopter waves and four are for surface waves. They are shown in Figure D.1 . The next section will describe each routine found in the model in greater detail.

E. DESCRIPTION OF ROUTINES

The following routines are listed in order of appearance in the program. When two routines appear separated by a slash, the first routine refers to helicopters and the second refers to surface. The reason for this format is due to the similarity in the logic of the routines. Flow diagrams of all the routines are held by the author if needed.

1. Routine Name: SETUP

a. Purpose

The initial portion of this routine reads values from the T-1.DAT file, interactive input, and input found within the code itself. These values are used to dimension arrays used for landing plan data and set up initial conditions for the simulation. SETUP then opens the landing plan file for input of the values contained in that file. The basic language reads each line from the file as a character string, proceeds to strip off individual pieces, converts those pieces into numerical values, and then assigns them to arrays for later use. This procedure continues until the end of the data file is reached at which time the file is closed.

Another purpose for this section is to normalize IHOUR, HHOUR, and CHOUR by setting the lowest time to zero. It then adjusts the other times to reflect their original differences between the variables.

b. Type function

This is a routine that is part of the main program. It is not a subroutine or other type of function.

c. Calling routines

None

d. Called routines

Calls INITIAL.

e. Files created or used

This routine uses two files for input: T-1.DAT and the landing plan file. It also creates a file for output of the time, number of personnel ashore, and firepower ashore which is used for graphical presentation.

f. Flow diagram

Available but not contained in this manual.

2. Routine Name: INITIAL

a. Purpose

This routine has several functions to perform within the model framework.

- It initializes the time sort arrays which are used to track the waves as they progress through the various activities.
- It initializes the deck spots and well deck spots for use by the load activity.

- It computes times for landing craft to launch from the ships.
- It ensures sufficient landing craft for scheduled and on-call waves are available in the landing plan file.
- It initializes the following arrays: SSH, SCH, SNA, SSL, SCL, and SNA. These arrays store the serial number and its corresponding wave number for each line of input from the landing plan data file. They are used throughout the program to keep track of all the serials in a particular wave.

b. Type function

This function is a subroutine.

c. Calling routines

INITIAL is called by SETUP routine.

d. Called routines

INITIAL calls HSSORT, HCSORT, LSSORT, LCSORT.

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

3. Routine Name: Wave time sort routines

HSSORT: Scheduled helicopter waves time sort

HCSORT: On-call helicopter waves time sort

HNSORT: Nonscheduled helicopter time sort

LSSORT: Scheduled surface waves time sort

LCSORT: On-call surface waves time sort

INSORT: Nonscheduled surface waves time sort

a. Purpose

The purpose of the time sort routine is to find the event with the lowest time amongst all waves within a particular class (e.g. HSSORT will search through the scheduled helicopter waves 'looking' for the lowest time for the next event). Once that wave has been identified, the information pertaining to its wave number, classification, time to next event, load factor, mode, etc. will be transferred to the MCLOCK array which stores the same information from the five other time sort routines. The MCLOCK array is then used by the MSORT routine .

b. Type function

Subroutine

c. Calling routines

Each activity subroutine calls the time sort routine according to the class of wave that it is processing at the time. For instance, HRLD (helicopter load routine), will call HSSORT, HCSORT, and HNSORT depending an the class of wave being processed.

d. Called routines

None

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

4. Routine Name: EXEC

a. Purpose

EXEC is the central control routine of the model. It controls the flow of all waves by calling the proper routine according to the class and mode of the wave which has the lowest time for the next event. EXEC first checks whether certain flags have been set or not. These flags or signals are concerned with special conditions such as lack of sufficient helicopters, lack of deck spots, etc. If the flags have been set, the proper subroutine will be called to make adjustments in the time accordingly. The subroutine MSORT is then called to provide the wave which has the lowest time for the next event. EXEC then routes that wave to the activity subroutine for processing.

b. Type function

This function is considered the main program.

c. Calling routines

None

d. Called routines

EXEC calls or can call the following routines: HCKLD, STNSH/STNSL, STOCH/STOCL, STSH/STSL, RESTART/RESTART, MSORT, HRLD/BRLD, HLNCH/BLNCH, HUNLD/BUNLD, HRTN/BRTN, HARIV/BARIV.

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

5. Routine Name: HRLD/BRLD

a. Purpose

This routine is designed to load (or attempt to load) a helicopter (surface) wave from available deck (well) spots on the ships. Loading will not occur if (1) there are no deck (well) spots or (2) there insufficient carriers of the proper type to carry the current wave.

b. Type function

Subroutine

c. Calling routines

These routines are called from EXEC.

d. Called routines

HRLD calls HSSORT, HCSORT, HNSORT, SPOTS and BRLD calls LSSORT, LCSORT, LNSORT, WELDEK.

e. Files created or used

None

f. Narrative

This subroutine is called from EXEC when MSORT indicates that the current wave scheduled for loading has the lowest time amongst all other waves. The subroutine first checks the class of the wave to be loaded and then the deck spot availability. If spots are not available the wave must wait for reactivation by RESTART (BESTART) as soon as the spots become available. If the spots are available, the routine compares the number of carriers and type that are needed with the number of that type in the carrier pools. If the number of carriers is insufficient to lift the wave, the wave sort array is advanced by 10 minutes to await return of sufficient carriers. If either insufficiency case occurs, the appropriate wave time sort routine is called to determine the next wave for processing.

For the case of enough carriers and spots to load the current wave, the wave parmeters are set for ready to load and launch. SPOTS is called to compute the time interval needed to load, DLTIME. Upon return to HRLD (BRLD) the time interval to load is added to the current time, HOUR, and the results are placed in the appropriate wave time sort array. This information is then sent through the appropriate wave time sort routine to determine the lowest time for the next event.

g. Flow diagram

See Figure D.2 .

6. Routine Name: SPOTS/WEIDEK

a. Purpose

This pair of routines determines the helo (well) deck spots that are available for use by the wave being processed by HRLD (BRLD). If the spots are available a flag is set and control is returned to the calling routine (HRLD/BRLD). If spots are available, the time necessary to load DLTIME, is computed then control is returned to the calling routine. DLTIME is a global variable and available throughout the program.

b. Type function

Subroutine

c. Calling routines

SPOTS/WEIDEK is called by HRLD/BRLD.

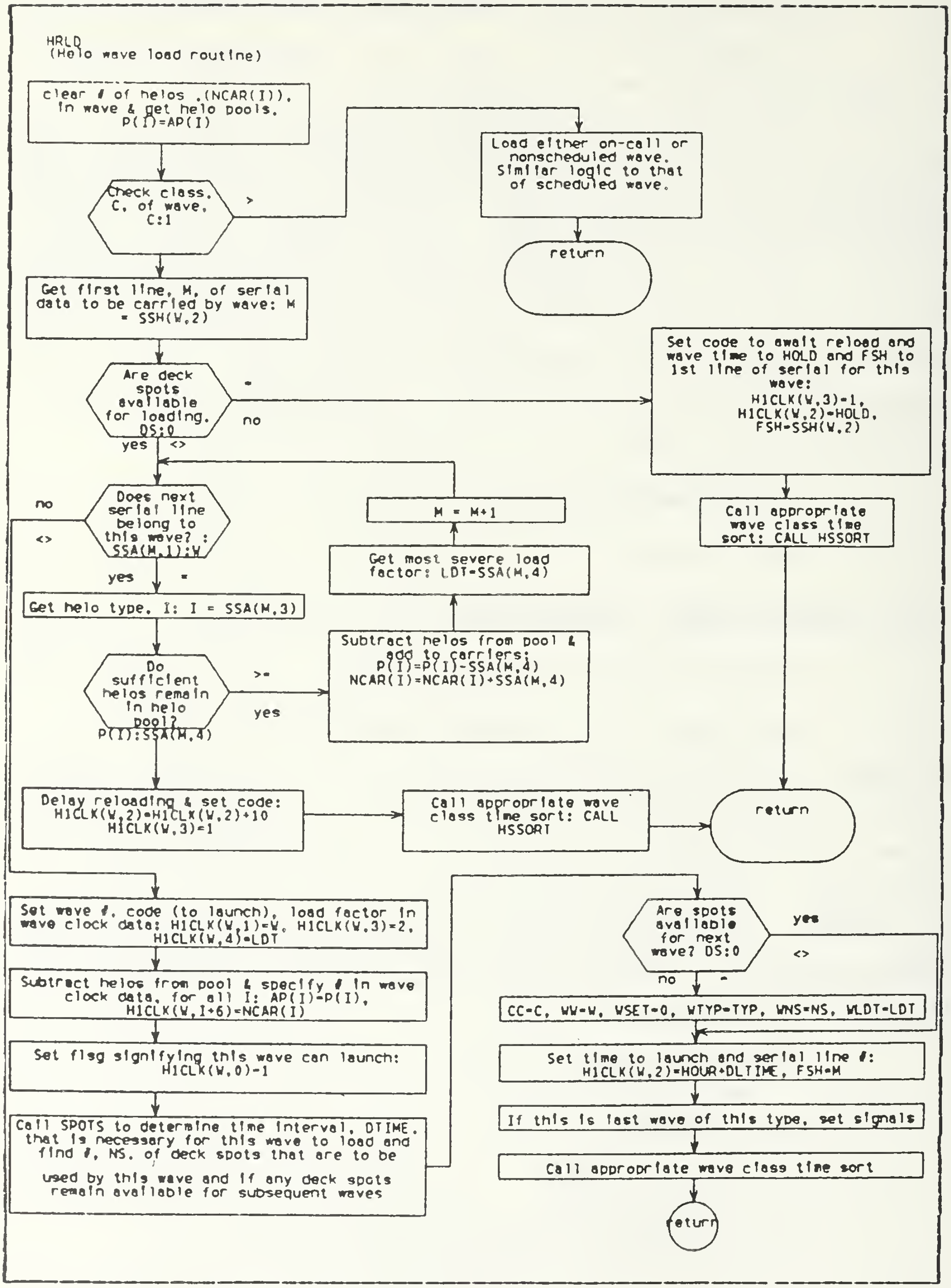

HRLD
(Helo wave load routine)
clear # of helos .(NCAR(I)), in wave & get helo pools, P(I)=AP(I)
Check class, C, of wave, C:1
>
Load either on-call or nonscheduled wave. Similar logic to that of scheduled wave.
return
Get first line, M, of serial data to be carried by wave: M = SSH(W,2)
Are deck spots available for loading, DS:0
=
no
yes
<>
Set code to await reload and wave time to HOLD and FSH to 1st line of serial for this wave:
H1CLK(W,3)=1,
H1CLK(W,2)=HOLD,
FSH=SSH(W,2)
no
<>
Does next serial line belong to this wave? : SSA(M,1):W
yes
=
M = M+1
Get most severe load factor: LDT=SSA(M,4)
Call appropriate wave class time sort: CALL HSSORT
Get helo type, I: I = SSA(M,3)
Do sufficient helos remain in helo pool? P(I):SSA(M,4)
>=
yes
Subtract helos from pool & add to carriers:
P(I)=P(I)-SSA(M,4)
NCAR(I)=NCAR(I)+SSA(M,4)
Delay reloading & set code:
H1CLK(W,2)=H1CLK(W,2)+10
H1CLK(W,3)=1
Call appropriate wave class time sort: CALL HSSORT
return
Set wave #, code (to launch), load factor in wave clock data: H1CLK(W,1)=W, H1CLK(W,3)=2, H1CLK(W,4)=LDT
Subtract helos from pool & specify # in wave clock data, for all I: AP(I)=P(I), H1CLK(W,I+6)=NCAR(I)
Set flag signifying this wave can launch: H1CLK(W,0)=1
Call SPOTS to determine time interval, DTIME, that is necessary for this wave to load and find #, NS, of deck spots that are to be used by this wave and if any deck spots remain available for subsequent waves
Are spots available for next wave? DS:0
yes
<>
no
=
CC=C, WW=W, WSET=0, WTYP=TYP, WNS=NS, WLDT=LDT
Set time to launch and serial line #: H1CLK(W,2)=HOUR+DLTIME, FSH=M
If this is last wave of this type, set signals
Call appropriate wave class time sort
return

Figure D.2 Flow Diagram of HRLD/BRLD.

d. Called routines

None

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

7. Routine Name: HLNCH/BLNCH

a. Purpose

These routines launch a wave from ships and determines the time it will arrive at the landing site. It determines the number of deck (helo or surface) spots made available to subsequent waves and restores those spots. Also it reevaluates the loading time for a wave being loaded which had employed all of the remaining deck spots.

b. Type function

Subroutine

c. Calling routines

These routines are called from EXEC.

d. Called routines

HLNCH calls HSSORT, HCSORT, HNSORT, HICDCK and BLNCH calls LSSORT, ICSORT, LNSORT.

e. Files created or used

None

f. Narrative

These routines are called from EXEC when the time for launching the current wave is lowest among all other classes of waves processed by MSORT. The subroutine first checks the class of the wave and then gets the first line of serials in that wave to be carried. The number and type of carrier is obtained from the wave time sort array. A launch signal is placed in the serial lines for this wave and the load factor is extracted from the wave time sort array. The time interval, DT, from launch to arrival at the LZ is computed using the various input data such as launch distance, carrier speed, load time, etc.

Before setting the appropriate wave time sort array (helos only) to the arrival time at the LZ, a check is made to see if all the deck spots are in use by the current wave. If so, then the launching of the current wave would make available a certain number of spots for the next wave. If not all the spots are in use, the routine HIODCK is called to reevaluate the helicopter deck spot employment to determine how many spots will be available after launch of the current wave.

Finally, the time to load is determined by adding DT to the current time, HOUR, and setting the proper code for launch. A call to the appropriate wave time sort routine follows these computations. Any deck spots used by the currnet wave are then restored upon launch.

g. Flow diagram

See Figure D.3 .

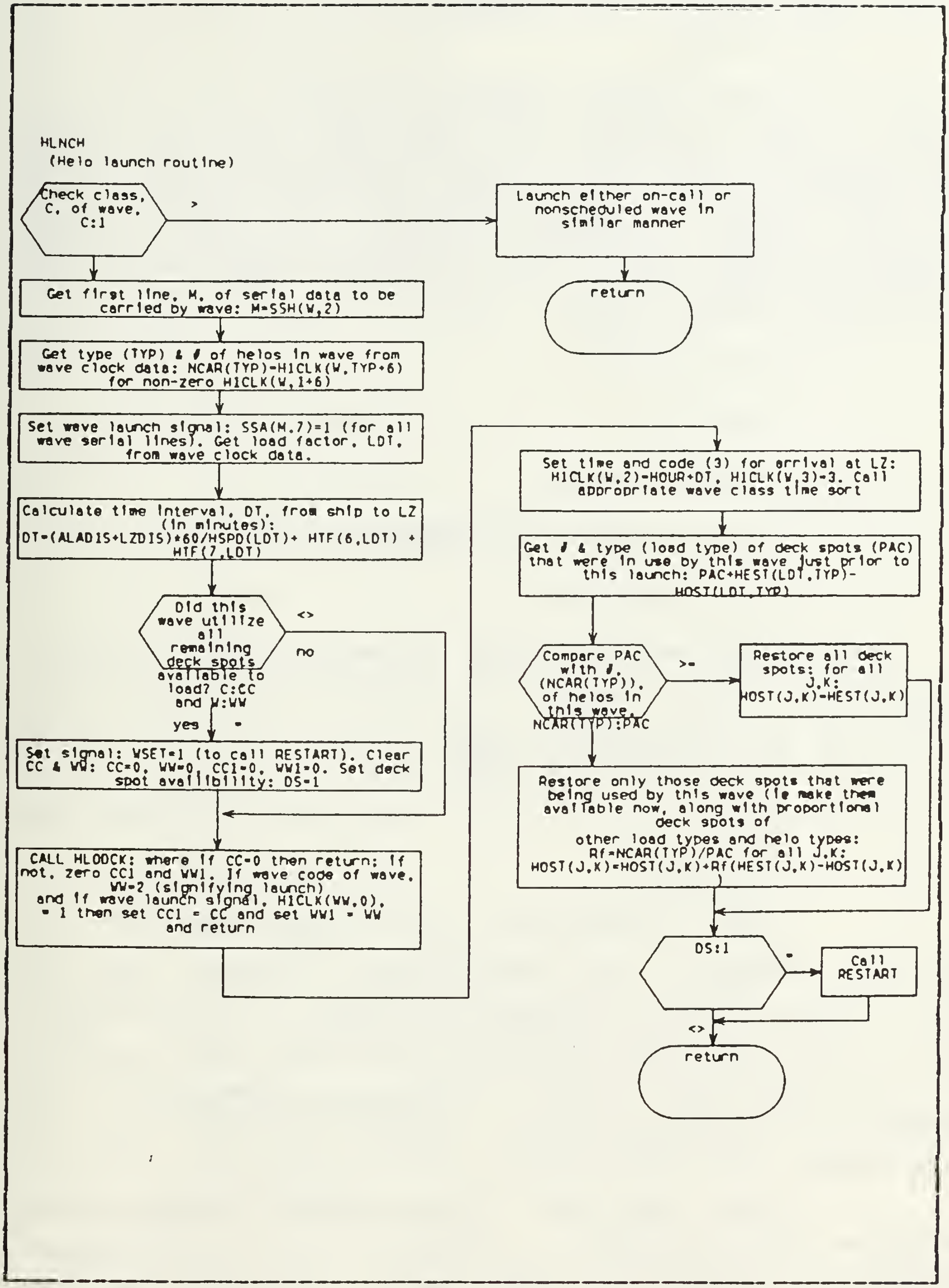

HLNCH
(Helo launch routine)
Check class, C, of wave, C:1
>
Launch either on-call or nonscheduled wave in similar manner
return
Get first line, M, of serial data to be carried by wave: M=SSH(W,2)
Get type (TYP) & # of helos in wave from wave clock data: NCAR(TYP)=HICLK(W,TYP+6) for non-zero HICLK(W,1+6)
Set wave launch signal: SSA(M,7)=1 (for all wave serial lines). Get load factor, LDT, from wave clock data.
Calculate time interval, DT, from ship to LZ (in minutes): DT=(ALADIS+LZDIS)*60/HSPD(LDT)+ HTF(6,LDT) + HTF(7,LDT)
Did this wave utilize all remaining deck spots available to load? C:CC and W:WW
<>
no
yes
=
Set signal: WSET=1 (to call RESTART). Clear CC & WW: CC=0, WW=0, CC1=0, WW1=0. Set deck spot availibility: DS=1
CALL HLODCK: where if CC=0 then return; if not, zero CC1 and WW1. If wave code of wave, WW=2 (signifying launch) and if wave launch signal, HICLK(WW,0), = 1 then set CC1 = CC and set WW1 = WW and return
Set time and code (3) for arrival at LZ: HICLK(W,2)=HOUR+DT, HICLK(W,3)=3. Call appropriate wave class time sort
Get # & type (load type) of deck spots (PAC) that were in use by this wave just prior to this launch: PAC=HEST(LDT,TYP)-HOST(LDT,TYP)
Compare PAC with #, (NCAR(TYP)), of helos in this wave, NCAR(TYP):PAC
>=
Restore all deck spots: for all J,K: HOST(J,K)=HEST(J,K)
Restore only those deck spots that were being used by this wave (ie make them available now, along with proportional deck spots of other load types and helo types: Rf=NCAR(TYP)/PAC for all J,K: HOST(J,K)=HOST(J,K)+Rf(HEST(J,K)-HOST(J,K))
DS:1
=
Call RESTART
<>
return

Figure D.3 Flow Diagram of HLNCH/BLNCH.

8. Routine Name: HUNLD/BUNLD

a. Purpose

These routines determine whether or not landing sites are still occupied by previous waves and if so, delays the landing of the current wave until the pertinent zones are free to use. Also it calculates the length of time necessary for the current wave to unload (constrained by the size of the landing zone).

b. Type function

Subroutine

c. Calling routines

These routines are called from EXEC.

d. Called routines

HUNLD calls HSSORT, HCSORT, HNSORT and BUNLD calls LSSORT, LCSORT, LCSORT.

e. Files created or used

None

f. Narrative

These routines are called from EXEC when the time for landing at the landing site is lowest among all other classes of waves processed by MSORT. The subroutine first checks the class of the wave and then gets the first line of serial data in that wave to be processed. The number and type of carrier is obtained from the wave time sort array.

Next, each line of serial data is examined to determine its destination. If one of the landing zones/beaches is occupied by a prior wave, the wave time sort array is set to await clearance of the landing site. Control then returns to EXEC.

If all of the zones are clear, HUNLD/BUNLD computes the land and unload time interval, ZTIME. That time is added to the current time, HOUR, the event code is set to "return to ship", and the appropriate wave time sort routine is called. Finally, the time for completion of the unload is computed and placed in an array which is checked by subsequent waves to determine occupancy of the landing site.

g. Flow diagram

See Figure D.4 .

9. Routine Name: HRTN/BRTN

a. Purpose

The purpose of these routines is to print a running total of the personnel ashore and the firepower ashore. Also HRTN calculates the time that the wave will return to the area of the shipping and hence be available for formation of subsequent waves.

b. Type function

Subroutine

c. Calling routines

These routines are called from EXEC.

d. Called routines

HRTN calls HSSORT, HCSORT, HNSORT and BRTN calls LSSORT, LCSORT, LNSORT.

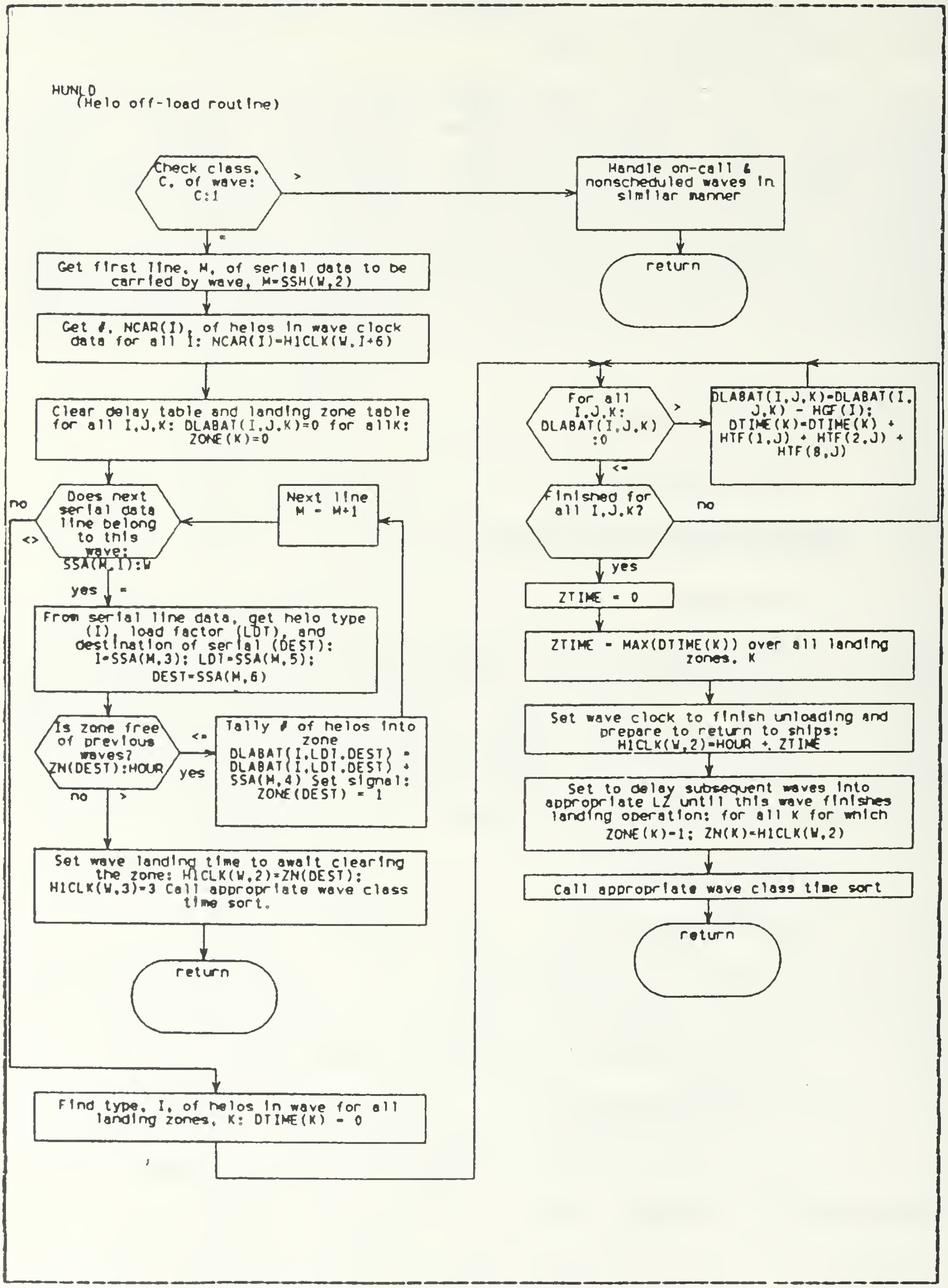

HUNLD
(Helo off-load routine)
Check class, C, of wave: C:1
>
=
Handle on-call & nonscheduled waves in similar manner
return
Get first line, M, of serial data to be carried by wave, M=SSH(W,2)
Get #, NCAR(I), of helos in wave clock data for all I: NCAR(I)=HICLK(W,I+6)
Clear delay table and landing zone table for all I,J,K: DLABAT(I,J,K)=0 for all K: ZONE(K)=0
Does next serial data line belong to this wave: SSA(M,1):W
no
<>
yes
=
Next line M = M+1
From serial line data, get helo type (I), load factor (LDT), and destination of serial (DEST): I=SSA(M,3); LDT=SSA(M,5); DEST=SSA(M,6)
Is zone free of previous waves? ZN(DEST):HOUR
<=
yes
no
>
Tally # of helos into zone DLABAT(I,LDT,DEST) = DLABAT(I,LDT,DEST) + SSA(M,4) Set signal: ZONE(DEST) = 1
Set wave landing time to await clearing the zone: HICLK(W,2)=ZN(DEST): HICLK(W,3)=3 Call appropriate wave class time sort.
return
Find type, I, of helos in wave for all landing zones, K: DTIME(K) = 0
For all I,J,K: DLABAT(I,J,K) :0
>
DLABAT(I,J,K)=DLABAT(I,J,K) - HGF(I); DTIME(K)=DTIME(K) + HTF(1,J) + HTF(2,J) + HTF(8,J)
<=
Finished for all I,J,K?
no
yes
ZTIME = 0
ZTIME = MAX(DTIME(K)) over all landing zones, K
Set wave clock to finish unloading and prepare to return to ships: HICLK(W,2)=HOUR + ZTIME
Set to delay subsequent waves into appropriate LZ until this wave finishes landing operation: for all K for which ZONE(K)=1; ZN(K)=HICLK(W,2)
Call appropriate wave class time sort
return

Figure D.4 Flow Diagram of HUNLD/BUNLD.

e. Files created or used

None

f. Narrative

These routines are called from EXEC when the time for unload completion is lowest among all other classes of waves processed by MSORT. The class of wave is checked and the time interval necessary to return to the ship is calculated.

The running totals of personnel and firepower ashore are tallied and sent to either the screen or the printer. The code indicating that unloading has been completed is set. Finally, the wave time sort array is updated with the time interval to arrive back at the ship and the appropriate wave class time sort subroutine is called.

g. Flow diagram

See Figure D.5 .

10. Routine Name: HARIV/BARIV

a. Purpose

These routines return the carriers to their respective pools for use in forming subsequent waves after degrading the number of helicopters to reflect attrition losses. It also initiates the loading process for subsequent waves.

b. Type function

Subroutine

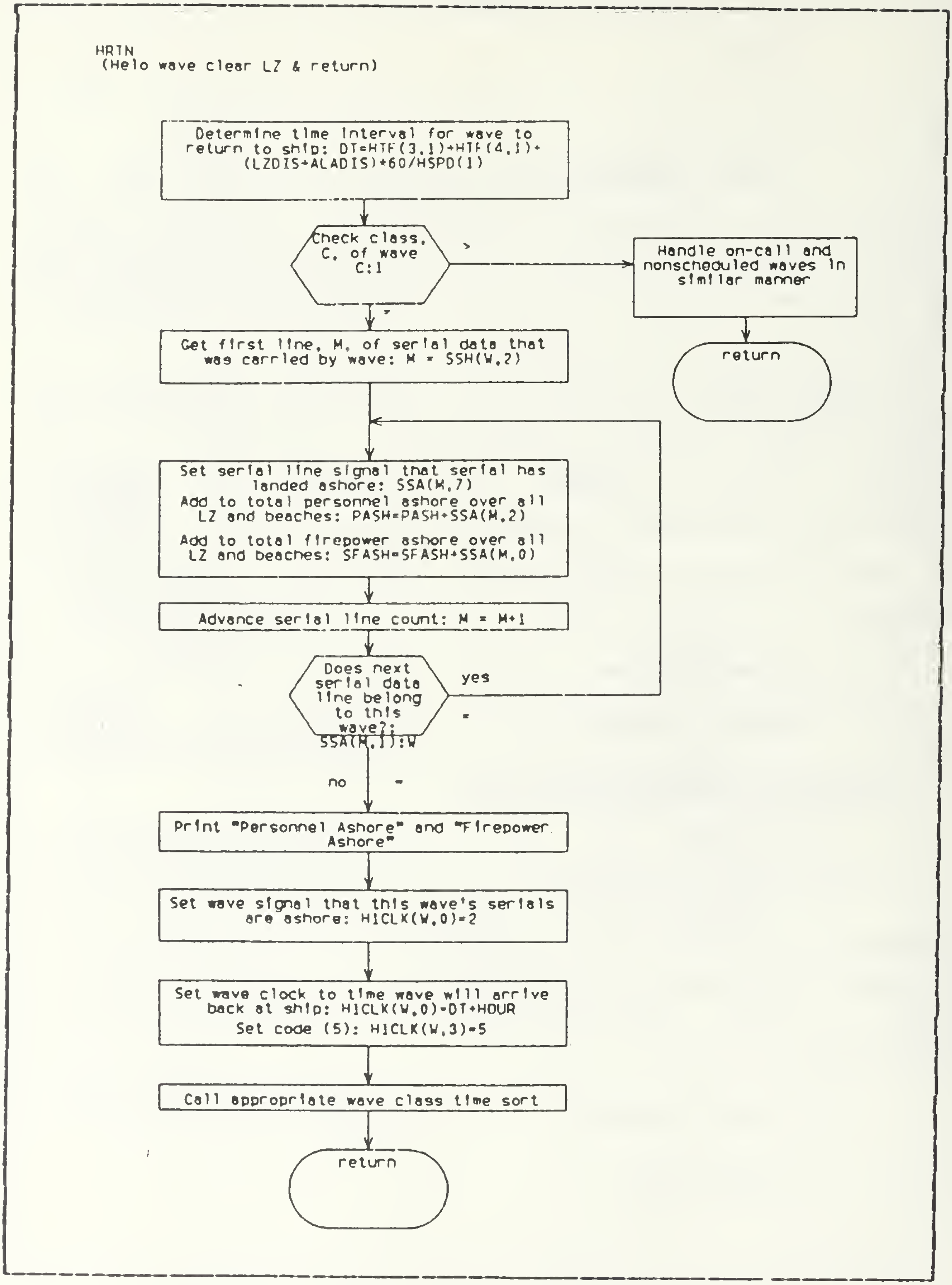

HRTN
(Helo wave clear LZ & return)
Determine time interval for wave to return to ship: DT=HTF(3,1)+HTF(4,1)+(LZDIS+ALADIS)*60/HSPD(1)
Check class, C, of wave C:1
>
=
Handle on-call and nonscheduled waves in similar manner
return
Get first line, M, of serial data that was carried by wave: M = SSH(W,2)
Set serial line signal that serial has landed ashore: SSA(M,7)
Add to total personnel ashore over all LZ and beaches: PASH=PASH+SSA(M,2)
Add to total firepower ashore over all LZ and beaches: SFASH=SFASH+SSA(M,0)
Advance serial line count: M = M+1
Does next serial data line belong to this wave?: SSA(M,1):W
yes
=
no
≠
Print "Personnel Ashore" and "Firepower Ashore"
Set wave signal that this wave's serials are ashore: HICLK(W,0)=2
Set wave clock to time wave will arrive back at ship: HICLK(W,0)=DT+HOUR
Set code (5): HICLK(W,3)=5
Call appropriate wave class time sort
return

Figure D.5 Flow Diagram of HRTN/BRTN.

c. Calling routines

These routines are called from EXEC.

d. Called routines

HARIV calls HSSORT, HCSORT, HNSORT and LARIV calls LSSORT, LCSORT, LNSORT.

e. Files created or used

None

f. Narrative

These routines are called from EXEC when the time for arrival back to the ship is the lowest among all other classes of waves processed by MSORT. The class of the wave is checked. If the returning wave is of the scheduled class, the number and type of carriers are placed in a carrier pool. At this point reduction of the returning carriers can be made by an attrition factor. Currently the model is set to place all of the returning carriers into the pool. However, by changing line 4310 (helo) or line 6660 (surface) in the program, the attrition can be set to whatever value desired.

Next, the returning wave time sort array is set to a high value to prevent that wave from being processed again. If the current wave is the last of the class, the appropriate wave time sort routine is called and control is returned to EXEC. Otherwise, the routine finds the next wave to load and sets the proper codes.

g. Flow diagram

See Figure D.6 .

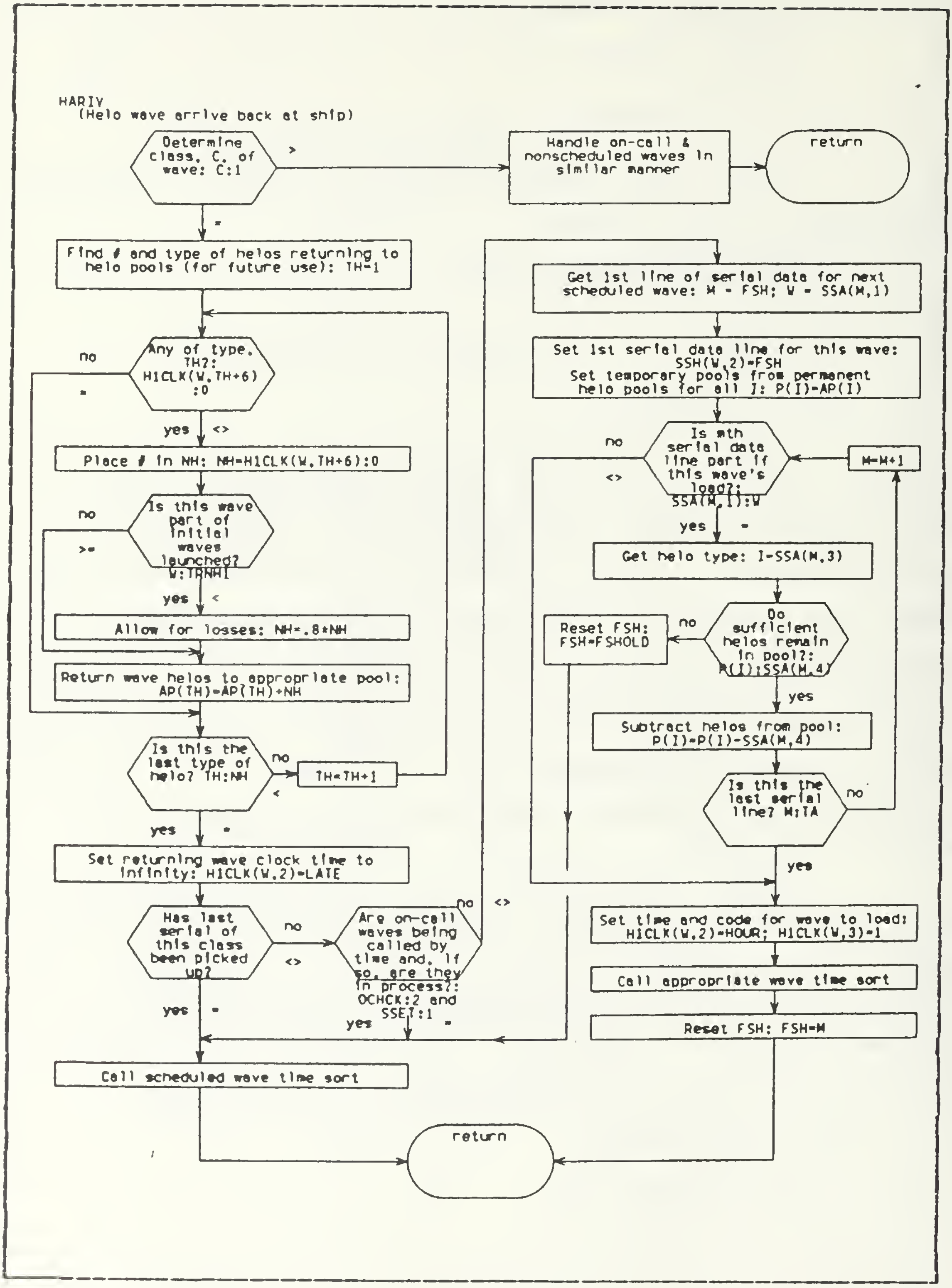

HARIV
(Helo wave arrive back at ship)
Determine class, C, of wave: C:1
>
Handle on-call & nonscheduled waves in similar manner
return
=
Find # and type of helos returning to helo pools (for future use): TH=1
Any of type, TH?: H1CLK(W,TH+6) :0
no
=
yes
<>
Place # in NH: NH=H1CLK(W,TH+6):0
Is this wave part of initial waves launched? W:TRNHI
no
>=
yes
<
Allow for losses: NH=.8*NH
Return wave helos to appropriate pool: AP(TH)=AP(TH)+NH
Is this the last type of helo? TH:NH
no
<
TH=TH+1
yes
=
Set returning wave clock time to infinity: H1CLK(W,2)=LATE
Has last serial of this class been picked up?
no
<>
yes
=
Are on-call waves being called by time and, if so, are they in process?: OCHCK:2 and SSET:1
no
<>
yes
=
Call scheduled wave time sort
return
Get 1st line of serial data for next scheduled wave: M = FSH; W = SSA(M,1)
Set 1st serial data line for this wave: SSH(W,2)=FSH
Set temporary pools from permanent helo pools for all I: P(I)=AP(I)
Is mth serial data line part if this wave's load?: SSA(M,1):W
no
<>
M=M+1
yes
=
Get helo type: I=SSA(M,3)
Do sufficient helos remain in pool?: P(I):SSA(M,4)
no
Reset FSH: FSH=FSHOLD
yes
Subtract helos from pool: P(I)=P(I)-SSA(M,4)
Is this the last serial line? M:TA
no
yes
Set time and code for wave to load: H1CLK(W,2)=HOUR; H1CLK(W,3)=1
Call appropriate wave time sort
Reset FSH: FSH=M

Figure D.6 Flow Diagram of HARIV/BARIV.

11. Routine Name: RESTART/BESTART

a. Purpose

This pair of routines is used to restart timing for waves stalled by insufficient helicopter deck spots or landing craft well deck spots. If there are not enough deck/well spots available, this routine adjusts the time to load by adding a time increment onto the wave sort array. Essentially, this action delays the loading until a reassessment determines the availibilty of spots for the wave which is next to load.

These routines also serve a second function and that is to add an increment of time to the wave sort arrays when switching from one class to another. This short delay is designed as a break in the process to simulate the shift from one class of wave to another.

b. Type function

Subroutine

c. Calling routines

This routine is called from EXEC.

d. Called routines

RESTART calls HSSORT, HCSORT, HNSORT and BESTART calls LSSORT, LCSORT, LNSORT.

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

12. Routine Name: STOCH/STCCL

a. Purpose

These routines initiate on-call waves to load after the final scheduled wave commenced to load. A signal is set by HRLD (ERLD) when the last scheduled helo/surface wave has begun the loading activity. This routine then sets up the on-call waves to begin the loading process. If the signal for on-call waves to interrupt the scheduled waves has been set, this routine would return immediately to EXEC and not be used.

b. Type function

Subroutine

c. Calling routines

These routines are called from EXEC.

d. Called routines

These routines call RESTART/BESTART.

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

13. Routine Name: STNSH/STNSL

a. Purpose

The purpose of these two subroutines is to start the nonscheduled helo/surface waves after the final on-call wave has commenced to load. A signal is set by HRLD (ERID) which tells the EXEC when the last on-call wave has started loading. EXEC then calls this routine to set up the nonscheduled waves for loading.

b. Type function.

Subroutine

c. Calling routines

These routines are called by EXEC.

d. Called routines

These routines call RESTART/BESTART.

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

14. Routine Name: STSH/STSL

a. Purpose

The function of these two routines is to restart scheduled waves that have been interrupted by on-call waves. If the flag is set to have on-call waves break into the scheduled wave sequence, the scheduled waves are temporarily halted by another routine. Once the designated on-call waves are complete, the scheduled wave sequence resumes by the calling of STSH/STSL.

b. Type function

Subroutine

c. Calling routines

These routines are called from EXEC.

d. Called routines

These routines call RESTART/BESTART.

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

15. Routine Name: HCKLD

a. Purpose

This routine reevaluates the helicopter deck spot employment if the current wave does not occupy all of the spots. By doing this the model can determine how many additional spots are available for the next wave. This will indicate that either the next wave can load or a delay will be encountered caused by a lack of deck spots. As soon as enough spots are available a flag will be set to initiate loading of the next wave.

b. Type function

Subroutine

c. Calling routines

This routine is called by EXEC.

d. Called routines

This routine calls WSPOT.

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

16. Routine Name: HLODCK

a. Purpose

HLODCK is a short routine which sets up additional signals required to call HCKLD from EXEC. Those signals indicate whether there are any available deck spots left after the current wave has loaded.

b. Type function

Subroutine

c. Calling routines

This routine is called from HLNCH

d. Called routines

None

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

17. Routine Name: WSPOT

a. Purpose

This routine calculates the number of helo deck spots to be used by the current wave and then how many remain for use by the next wave. This provides information required by HCKLD to determine whether the next wave can load or has to delay until the current wave launches.

b. Type function

Subroutine

c. Calling routines

This routine is called by HCKLD.

d. Called routines

None

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

18. Routine Name: HALTSH/HALTSL

a. Purpose

These routines will stall the remaining scheduled waves when they are interupted by on-call waves. If the flag is set to interrupt scheduled waves, this routine is called prior to commencement of the on-call waves. Then once the on-call waves designated to interrupt have completed loading, a flag will be set to call STSH/STSL which restarts the stalled waves.

b. Type function

Subroutine

c. Calling routines

This routine is called from EXEC.

d. Called routines

None

e. Files created or used

None

f. Flow diagram

Available but not contained in this manual.

F. DATA BASE DESCRIPTION

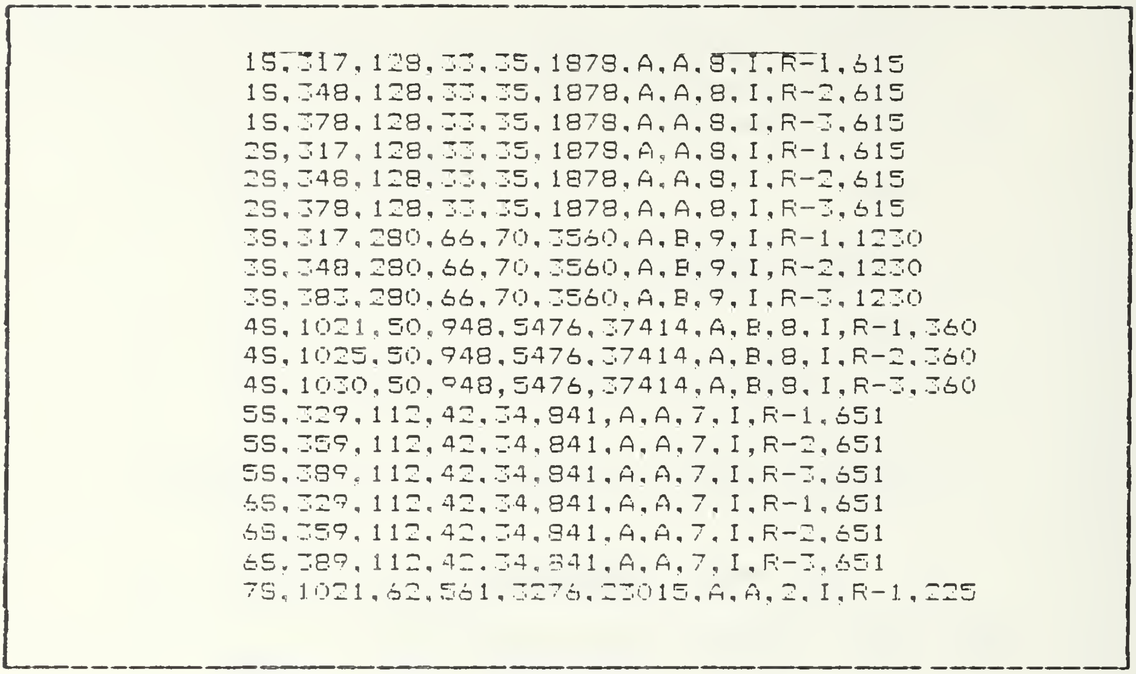
1. File name: Landing Plan data file

a. Purpose

Input data is broken down into three categories. The largest group of data is found in the landing plan file. An example of this file is found in D.7 . The landing plan file contains the wave and serial data which is processed by the model. It has type of carriers, number of personnel, wave number, destination, etc. for each serial going ashore. Its purpose is to provide the model with the troop unit, equipment, and transportation information which will be 'moved ashore' by the simulation.

b. Format

The landing plan file requires the manual construction of serials and assault waves in a specific format. Each line in the data file contains twelve pieces of information which the model requires. These are explained as part of an example listed in the 'help' file under landing plan file (Section B). This information is



```
1S,317,128,33,35,1878,A,A,8,I,R-1,615
1S,348,128,33,35,1878,A,A,8,I,R-2,615
1S,378,128,33,35,1878,A,A,8,I,R-3,615
2S,317,128,33,35,1878,A,A,8,I,R-1,615
2S,348,128,33,35,1878,A,A,8,I,R-2,615
2S,378,128,33,35,1878,A,A,8,I,R-3,615
3S,317,280,66,70,3560,A,B,9,I,R-1,1230
3S,348,280,66,70,3560,A,B,9,I,R-2,1230
3S,383,280,66,70,3560,A,B,9,I,R-3,1230
4S,1021,50,948,5476,37414,A,B,8,I,R-1,360
4S,1025,50,948,5476,37414,A,B,8,I,R-2,360
4S,1030,50,948,5476,37414,A,B,8,I,R-3,360
5S,329,112,42,34,841,A,A,7,I,R-1,651
5S,359,112,42,34,841,A,A,7,I,R-2,651
5S,389,112,42,34,841,A,A,7,I,R-3,651
6S,329,112,42,34,841,A,A,7,I,R-1,651
6S,359,112,42,34,841,A,A,7,I,R-2,651
6S,389,112,42,34,841,A,A,7,I,R-3,651
7S,1021,62,561,3276,23015,A,A,2,I,R-1,225
```

Figure D.7 Example of Landing Plan Data File.

also located on the same disk as the main program and is for use in constructing the data file. A copy of the 'help' file is located in Appendix E.

c. Routines

This file is called by the SETUP routine.

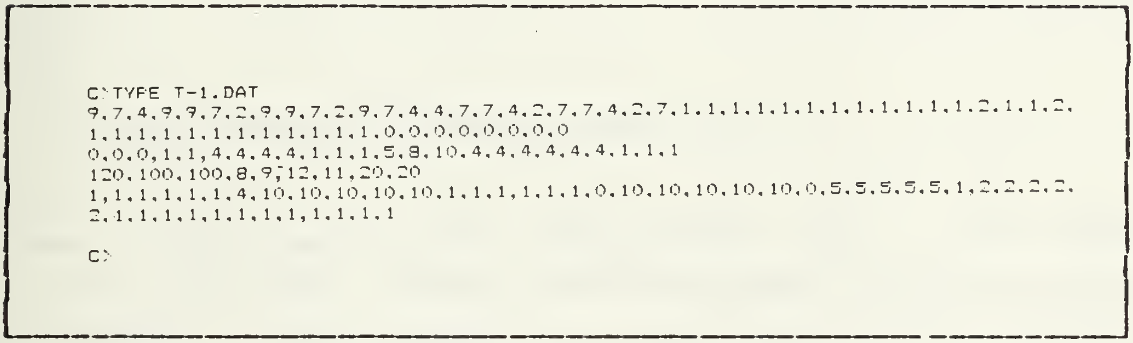
d. Updating

Each time a new landing plan is used, all the parameters pertaining to the number of waves, the number of landing craft for scheduled and on-call waves, the name of the landing plan file, etc. must be changed to conform to the new data. If this is not accomplished an error will occur identifying the line in the code where an inconsistency was found. Most of those values are found in the interactive mode which helps to prevent those type of errors.

2. File name: T-1.DAT

a. Purpose

This data file contains parameter values which are not subject to constant change (see Figure D.8). The three major tables of this type are the helicopter time factors, boat time factors, and operational deck spot tables. The first two provide the expected times for various evolutions to utilize and the last gives the deck spots under various conditions. Tables XXII, XXIII, and XXIV list the data which are read from T-1.DAT. The purpose of this file is to read data into various arrays within the program which will be used to compute event times.



```
C>TYPE T-1.DAT
9,7,4,9,9,7,2,9,9,7,2,9,7,4,4,7,7,4,2,7,7,4,2,7,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,2,
1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0
0,0,0,1,1,4,4,4,4,1,1,1,5,8,10,4,4,4,4,4,4,1,1,1
120,100,100,8,9;12,11,20,20
1,1,1,1,1,1,4,10,10,10,10,10,1,1,1,1,1,1,0,10,10,10,10,10,0,5,5,5,5,5,1,2,2,2,2,
2,1,1,1,1,1,1,1,1,1,1,1,1

C>
```

Figure D.8 Example of Data File T-1.DAT.

b. Format

The first two lines of the T-1.DAT file refer to Table XXIV . The data is listed in the file row by row from the table. Thus, the first twelve values come from row one of the table. The next line contains data from table XXII . It is listed in the same manner as Table XXIV except the first four values come from the first row of the table. Line four lists three helicopter speeds then six landing

TABLE XXII

Helicopter Time Factors in Minutes

Helo wave operation	Type of Load Internal	 External	 Combined
(1) Land in zone	0	0	0
(2) Disembark troops (cargo)	1	1	4
(3) Takeoff/rendezvous from LZ	4	4	4
(4) Maneuver/land ship	1	1	1
(5) Refuel/embark troops (cargo)	5	8	10
(6) Takeoff/rendezvous from ship	4	4	4
(7) Maneuver/land in LZ	4	4	4
(8) Safety clearance time	1	1	1

craft speeds. The vehicles for which these speeds are listed are: CH-53, CH-46, UH1N, LVT, LCM-6, LCM-8, LCU, LST, and LARC. The last two lines of the file contain data from Table XXIII They are also listed by row. Section C of Appendix E contains the format for the T-1.DAT file.

c. Routines

This file is called by the SETUP routine.

d. Updating

Updating is not necessary unless different time factors are desired for use.

TABLE XXIII

Boat Time Factors in Minutes

	Boat type						
Boat wave operation	LVT	LCM-6	LCM-8	LCU	LST	LARC	LCAC
(1) Land at beach (or LZ)	1	1	1	1	1	1	1
(2) Disembark troops/load	4	10	10	10	10	10	15
(3) Rendezvous for return	1	1	1	1	1	1	0
(4) Dock time to ship (well deck)	0	10	10	10	10	10	10
(5) Refuel/embark troops/load	0	5	5	5	5	5	15
(6) Circle and rendezvous	1	2	2	2	2	2	1
(7) Land at beach	1	1	1	1	1	1	1
(8) Safety clearance time	1	1	1	1	1	1	1

3. File Name: Interactive/line input

This data source is found within the program itself as either a line of code or as an interactive query. The data is of the type that may be changed easily for various types of analysis. Although this source is not a file, it is the final means used to assign values to the remaining variables. Once the first two files are prepared, the model is loaded into the computer and initiated. The program will query the user to input data for the parameters which are most likely to change during any type of analysis. An alternative is to change the code within the program itself where some of these values are found.

a. Format

Figure D.9 contains an example of the interactive input while the parameters that are requested are listed in the Table XXV .

TABLE XXIV

Operational Deck Spots

	Internal Loading				External Loading				Combination Loading			
Ship type	A	B	C	D	A	B	C	D	A	B	C	D
LHA(#1)	9	7	4	9	9	7	2	9	9	7	2	9
LPH(#2)	7	4	4	7	7	4	2	7	7	4	2	7
LSD(#3)	1	1	1	1	1	1	1	1	1	1	1	1
LPD(#4)	2	1	1	2	1	1	1	1	1	1	1	1
LST(#5)	1	1	1	1	0	0	0	0	0	0	0	0

Helo types: A=CH46E B=CH53D C=CH53E D=UH1N

b. Routines

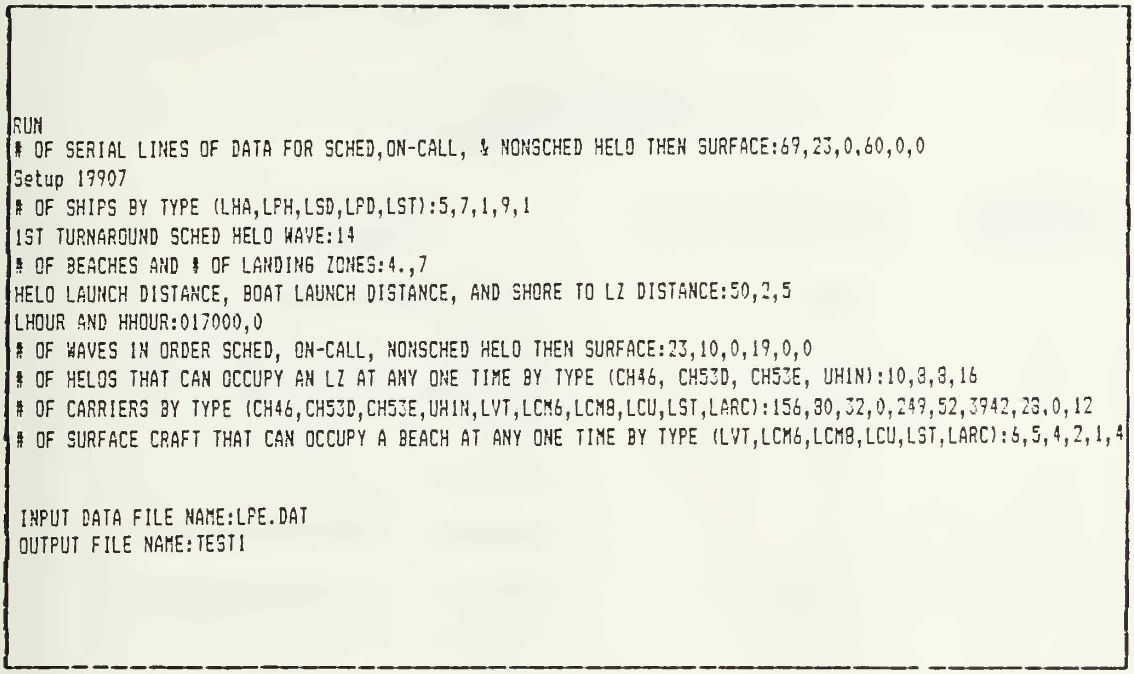
None

c. Updating

Updating of this input data is relatively simple. The majority of parameters which would need to be changed will be requested by the interactive queries. Others can be changed by a direct overwrite in the program listing.

G. SOURCE LISTING

See Appendix F.



```
RUN
# OF SERIAL LINES OF DATA FOR SCHED,ON-CALL, & NONSCHED HELO THEN SURFACE:69,23,0.60,0,0
Setup 19907
# OF SHIPS BY TYPE (LHA,LPH,LSD,LPD,LST):5,7,1,9,1
1ST TURNAROUND SCHED HELO WAVE:14
# OF BEACHES AND # OF LANDING ZONES:4.,7
HELO LAUNCH DISTANCE, BOAT LAUNCH DISTANCE, AND SHORE TO LZ DISTANCE:50,2,5
LHOUR AND HHOUR:017000,0
# OF WAVES IN ORDER SCHED, ON-CALL, NONSCHED HELO THEN SURFACE:23,10,0,19,0,0
# OF HELOS THAT CAN OCCUPY AN LZ AT ANY ONE TIME BY TYPE (CH46, CH53D, CH53E, UH1N):10,8,8,16
# OF CARRIERS BY TYPE (CH46,CH53D,CH53E,UH1N,LVT,LCM6,LCM8,LCU,LST,LARC):156,80,32,0,249,52,3942,28.0,12
# OF SURFACE CRAFT THAT CAN OCCUPY A BEACH AT ANY ONE TIME BY TYPE (LVT,LCM6,LCM8,LCU,LST,LARC):6,5,4,2,1,4


 INPUT DATA FILE NAME:LPE.DAT
 OUTPUT FILE NAME:TEST1
```

Figure D.9 Example of Interactive/Line Input Data.

H. GLOSSARY OF VARIABLES

See Section F of Appendix E.

I. MODEL TEST RESULTS

See Chapter V of this thesis.

TABLE XXV

Interactive Parameters

Order of Request	Parameter Name	Definition
1	TA	Sched helo serial #
2	CA	On-call helo serial #
3	NA	Nonsched helo serial #
4	TS	Sched surf serial #
5	CS	On-call surf serial #
6	MS	Nonsched surf serial #
7-12	NUS(I)	Ship type (five types)
13	TRNH1	1st turnaround wave
14	NNB	Number of beaches
15	NLZ	Number of LZs
16	ASADIS	Helo launch distance
17	LODDIS	LVT launch distance
18	LZDIS	Shore to LZ distance
19	HSWAVE	Sched wave #
20	HCWAVE	On-call wave #
21	HNWAVE	Nonsched wave #
22	LSWAVE	Sched wave #
23	LCWAVE	On-call wave #
24	LNWAVE	Nonsched wave #
25-28	HGF(I)	# helos max in LZ by type (4 types)
29-38	AT(I)	# carriers by type (10 types)
39-44	LGF(I)	# boats max at beach by type (6 types)

APPENDIX E

'HELP' FILE

A. TABLE OF CONTENTS

B 'Help' File Listing 166

C. Landing Plan File 167

D. T-1.DAT File 168

E. Input Parameters 169

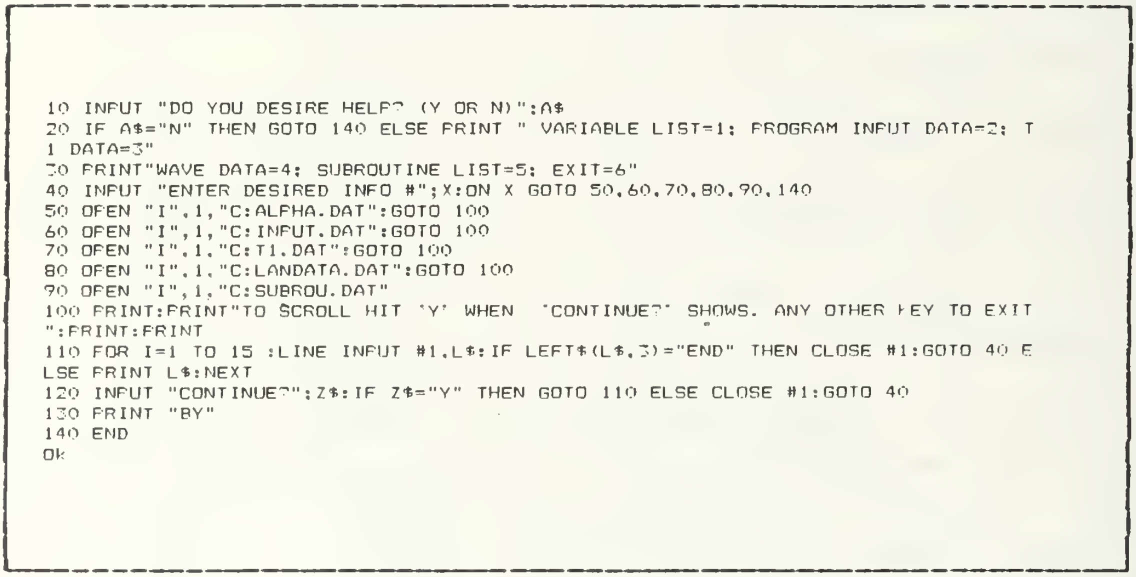
F. Subroutines 172

G. Variable List 174

H. Wave Clock Parameters 180

B. GENERAL

The 'help' file was developed to assist a user of the model with questions that will probably arise during operation. One can find the information while still on the computer without having to rely on a manual. A listing of the 'help' file is found in Figure A.1 .



```
10 INPUT "DO YOU DESIRE HELP? (Y OR N)":A$
20 IF A$="N" THEN GOTO 140 ELSE PRINT " VARIABLE LIST=1: PROGRAM INPUT DATA=2: T
1 DATA=3"
30 PRINT"WAVE DATA=4: SUBROUTINE LIST=5: EXIT=6"
40 INPUT "ENTER DESIRED INFO #";X:ON X GOTO 50,60,70,80,90,140
50 OPEN "I",1,"C:ALPHA.DAT":GOTO 100
60 OPEN "I",1,"C:INPUT.DAT":GOTO 100
70 OPEN "I",1,"C:T1.DAT":GOTO 100
80 OPEN "I",1,"C:LANDATA.DAT":GOTO 100
90 OPEN "I",1,"C:SUBROU.DAT"
100 PRINT:PRINT"TO SCROLL HIT 'Y' WHEN  'CONTINUE?' SHOWS. ANY OTHER KEY TO EXIT
":PRINT:PRINT
110 FOR I=1 TO 15 :LINE INPUT #1,L$:IF LEFT$(L$,3)="END" THEN CLOSE #1:GOTO 40 E
LSE PRINT L$:NEXT
120 INPUT "CONTINUE?";Z$:IF Z$="Y" THEN GOTO 110 ELSE CLOSE #1:GOTO 40
130 PRINT "BY"
140 END
Ok
```

Figure E.1 Help File Listing.

C. LANDING PLAN FILE

```
 type land the

      THE FOLLOWING IS AN EXAMPLE OF THE DATA INPUT FOR ALL WAVE
INFORMATION.  THE FIRST LINE OF LETTERS AND NUMBERS IS AN ACTUAL LINE OF
DATA FOUND IN THE LPE.DAT FILE. THE CORRESPONDING LETTERS WILL EXPLAIN
EACH COLUMN OF DATA.
     ................................................................
                  1S,317,128,33,35,1878,A,A,8,I,R-1,615
                  :   :   :  :  :   :   : : : :  :   :
                  A , B , C ,D ,E , F  ,G,H,I,J, K , L
     ................................................................

A:      THE NUMBER INDICATES THE WAVE # AND THE LETTER REFERS TO THE
        CLASS OF THE WAVE (S=SCHEDULED, O=ON-CALL, N=NON-SCHED).MORE
        THAN ONE LINE OF DATA (CALLED A SERIAL) CAN BE OF THE SAME WAVE
        NUMBER.  EACH SERIAL RUNS SEQUENTIALLY FROM 1 TO THE LAST SERIAL
        THE CLASS.

B:      RESERVED FOR FUTURE USE.

C:      NUMBER OF PERSONNEL IN PARTICULAR SERIAL OF A PARTICULAR WAVE

D-F:    RESERVED FOR FUTURE USE.

G:      MODE OF TRANSPORTATION. (A=HELO, S=SURFACE)

H:      CATEGORIES OF CARRIERS. A THROUGH K CORRESPOND TO 1-11. (1=CH46E
        2=CH53D, 3=CH53E, 4=UHIN, 5=LVT, 6=LCM-6, 7=LCM-8, 8=LCU, 9=LST,
        10=LARC, 11=LCAC). CARRIER TYPE MUST BE THE SAME FOR THE SAME
        WAVE.

I:      NUMBER OF CARRIERS IN THE SERIAL

J:      TYPE LOAD (I=INTERNAL, E=EXTERNAL, C=COMBINATION)

K:      DESTINATION: (B-1 THROUGH  B-4 REFERS TO BEACHES #1 THROUGH #4
        AND R-1 THROUGH R-4 REFERS TO LANDING ZONES #1 THROUGH #4)

L:      WEAPONS UNIT VALUE (WUV-FIREPOWER SCORE)
END OF FILE
```

D. T-1.DAT FILE

```
       THIS HELP FILE COVERS THE INPUT FILE CALLED DATA.T-1. THE
INFORMATION CONTAINED IN DATA.T-1 DEALS WITH PERMANENT VALUES SUCH AS
THE HELO TIME FACTORS, BOAT TIME FACTORS, ETC. THE DATA INPUT WILL BE
EXPLAINED IN THE ORDER IT IS CALLED FOR BY THE PROGRAM.

       ###############################################################

PROGRAM  VARIABLE    DEFINITION            LOCATION IN DATA.T-I
LINE #
-------  --------    ----------            --------------------
 60     HOS(I,J,K)   HELO DECK SPOTS       LINES I & 2

 70     HTF(I,J)     HELO TIME FACTORS     LINE 3

 90     HSPD(J)      HELO SPEEDS           LINE 4 (1ST 3 DATA POINTS)

 100    BSPD(I)      BOAT SPEEDS           LINE 4 (DATA POINTS 4-9)

 220    LTF(I,J)     BOAT TIME FACTORS     LINES 5 & 6

GENERAL INFO:
------- ----
        THE DATA IN PROGRAM LINE 60 IS LISTED ROW BY ROW FROM THE HELO
DECK TABLE FOUND IN APPENDIX B.  THE SUBSCRIPTS REFER TO VALUES AS
FOLLOWS: I=TYPE SHIP, J=TYPE LOAD, K=TYPE HELO.  THE FIRST TWELVE VALUES
IN THE DATA FILE COMES FROM THE FIRST LINE OF  VALUES IN THE TABLE. THE
SECOND TWELVE FROM THE SECOND ROW AND SO ON.
        PROGRAM LINE 70 INPUTS DATA FROM THE HELO TIME FACTORS TABLE
FOUND IN APPENDIX B.  THE SUBSCRIPT I REFERS TO THE TYPE OF OPERATION
AND J REFERS TO THE TYPE LOAD.  THE FIRST ROW OF THE TABLE FORMS THE
FIRST THREE VALUES IN THE DATA FILE.  THE SECOND THREE THE SECOND ROW
AND SO ON.
        HELO SPEED VALUES ARE FOUND IN LINE 90 OF THE PROGRAM.  THE
SUBSCRIPT REFERS TO THE GENERAL TYPE OF HELO.  J=I IS A CH-53, J=2 IS
A CH-46, AND J=3 IS A UH1N.  THE SECOND GROUP OF DATA FOUND IN LINE 100
IS THE BOAT SPEEDS.  THE SUBSCRIPT REFERS TO THE TYPE OF BOAT.  I=I IS
AN LVT, I=2 IS AN LCM-6, I=3 IS AN LCM-8, I=4 IS AN LCU, I=5 IS AN LST,
AND I=6 IS A LARC.
        PROGRAM LINE 220 INPUTS THE BOAT TIME FACTORS.  THE SUBSCRIPT I
REFERS TO THE TYPE OF BOAT WAVE OPERATION AND J REFERS TO THE TYPE OF
BOAT.  THE DATA FOUND IN THE FILE IS LISTED ROW BY ROW FROM THE TABLE.
THUS, THE FIRST ROW IN THE TABLE IS THE FIRST FOUR VALUES IN THE DATA
FILE AND SO ON.

END OF FILE
```

E. INPUT PARAMETERS

The following table identifies the input parameters, line number location, and a brief description of each variable. these variables must be initiated prior to running the model.

Variable Name	Line No.	Definiton
ALADIS	102	Distance from ship to shore for helos (nm)
ASADIS	102	Distance from ship to shore for LCAC (nm)
AT(I)	160	Initial # of carriers that will participate in landing (I=type carrier)
BSPD(I)	100	Boat speed (I=type boat: 1=LVT, 2=LCM-6, 3=LCM-8, 4=LCU, 5=LST, 6=LARC)
CA	40	Total # of lines of serial data for all on-call helo waves
CHOUR	112	Time for starting first LCAC wave
CLZDIS	102	Distance from shoreline to lcac LZ (nm)
CS	40	Total # of lines of serial data for all on-call surf waves
CSPD	102	LCAC speed
DLADIS	102	Distance boat must travel from ship to shore (nm)
HCWAVE	110	Total # on-call helo waves
HGF(I)	142	# helos by type that can land and unload simultaneously in LZ (I=helo type)
HHOUR	106	Time for starting first helo wave
HI	112	Very large value to prevent info to be considered in time sort (32000)
HNWAVE	110	Total # non-sched helo waves
HOLD	112	Same as HI (=16000)

HOS(I,J,K)	60	Operational deck spots for helo ops (I=ship type, J=type loading: internal,external, combination; K=type helo)
HSPD(I)	90	Helo speeds (I=type load: 1=internal, 2=external, 3=combination)
HSWAVE	110	Total # sched helo waves
HTF(I)	70	Table of helo time factors (see separate table)
LCWAVE	110	Total # of on-call boat waves
LGF(I)	190	# of boats by type that can unload simultaneously at numbered beach or LCAC LZ
LHOUR	106	Time for starting first boat wave (excluding LCACs)
LNWAVE	110	Total # of non-sched boat waves
LSWAVE	110	Total # of sched boat waves
LTF(I)	220	Boat time factors (I=boat wave operation, J=type boat)
LZDIS	102	Distance from shoreline to helo LZ (nm)
MS	40	Total # of lines of serial data for non-sched surf waves
NA	40	Total # of lines of serial data for non-sched helo waves
NBT	80	# boat types (5)
NCLZ	80	# of lcac lz's available to current landing plan
NCT	80	# lcac types (1)
NHT	80	# of helo types(4)
NLT	80	# of LVT types (1)
NLZ	80	# of lz's available to current landing plan
NNB	80	# of numbered beaches available to current landing plan
NST	80	# ships total from which can off load

NUS(I)	62	# of ships by type from which off-loading occurs (I=type of ship, 1=LHA; 2=LPH, 3=LSD, 4=LPD, 5=LST)
OCHCK	144	Signal identifying whether on-call helos are to follow scheduled (OCHCK =0) or break into sched wave sequence (OCHCK=2). If =2 then times must be specified for each on-call wave to commence operation
OCHTAB(I)	144	Time for on-call helo wave to commence operation if not following sequence (I=wave #). scheduled waves are delayed until on-call operations completed
OCLCK	144	Signal identifying whether on-call boats are to follow scheduled (OCLCK =0) or break into sched wave sequence (OCLCK=2). If =2 then times must be specified for each on-call wave to commence operation
SSPD(I)	102	Ship speed (I=type ship: 1=LHA, 2=LPH, 3=LSD, 4=LPD, 5=LST)
TA	40	Total # lines of serialized data for sched helo waves
TRNH1	80	The # of the first scheduled helo wave in first turnaround wave
TRNS1	80	The # of the first scheduled surf wave in first turnaround wave
TS	40	Total # lines of serialized data for sched surf waves
WDS(I)	144	Well deck spot array (I=type ship)

F. SUBROUTINES

This file lists all of the subroutines within the program code and their location by line number. The short description gives a general summary of what the routine does. More detailed desciptions of the major routines are contained in the model documentation.

Routine Name	Line No.	Description
BARIV	6630	Arrive back ship-surf
BESTART	7930	Restart timing for surf waves stalled for insufficient well deck spots
BLNCH	5400	Launch-surf
BRLD	4990	Loading-surf
BRTN	6160	Deposit cargo-surf
BUNLD	5790	Arrive/unload lz-surf
EXEC	2270	Controls subroutines and timing of model
HALTSH	7900	Stall remaining sched when on-call told to start before completion of sched air
HALTSL	8360	Stall remaining sched when on-call told to start before completion of sched surf
HARIV	4270	Arrive back at ship-air
HCKLD	8390	Reevaluates deck spot employment if current wave does not occupy all spots
HCSORT	3480	Sort routine for on-call air
HLNCH	3700	Launch routine for air
HLODCK	8640	Sets up recalculation and signals to call hckld from exec called from HLNCH
HNSORT	3550	Sort routine for non-sched waves-air
HRLD	2550	Load subroutine for air

HRTN	4610	Deposit cargo subroutine-air
HSSORT	3400	Sort routine for sched air
HUNLD	4060	Unload at lz subroutine
INITIAL	470	Initializes all arrays and sort routines
LCSORT	4860	Sort routine for on-call surf
LNSORT	4920	Sort routine for non-sched surf
LSSORT	4800	Sort routine for sched surf
MSORT	3630	Master sort routine-computes next event from all other sort routines
RESTART	7080	Restart timing for waves stalled by insufficient helo deck spots
SETUP	10	Initial input of data from data files and within program
SPOTS	3200	Calculates # of spots to be used-called by HRLD
STOCH	7750	Start on-call helo after final scheduled wave began to load
STOCL	7750	Start on-call surface after final scheduled wave began to load
STSH	7860	Restart scheduled waves after stall in HSLTSH (air)
STSL	8320	Restart scheduled waves after stall in HALTSL (surf)
STNSH	7820	Start non-scheduled helo after final on-call wave began to load
STNSL	8280	Start non-scheduled surface waves after final on-call wave began to load
WELDEK	5320	Calculates # spots to be used for surface craft (called by BRLD)
WSPOT	8710	Calculates # spots to be used by air (called by HCKLD)

G. VARIABLE LIST

The following table provides a listing of the program variables and their meanings. It contains all the major variables and most of the lesser variables.

A	Activity: helo (1=load,2=launch,3=arrive LZ,4=deposit cargo,5=arrive back ship), surface (1=launch preboat,2=arrive beach,3=deposit cargo,4=arrive back ship(except lvt's),5=reload from well-deck)
ACODE	Code for next event: 1=start loading wave,2=launch loaded wave,3=reach destination(LZ),4=complete unloading and start return trip,5=arrive back ship
ALADIS	Distance from ship to shore for helos(nm)
AP(I)	# of carriers in pool for future use (I=type carrier)
ASADIS	Distance from ship to shore for LCAC(nm)
AT(I)	Initial # of carriers that will participate in landing (I=type carrier)
BEGNSH	=1 is flag to begin non-sched helo wave (subroutine EXEC)
BEGOCH	=1 is flag to begin on-call helo wave (subroutine EXEC)
BEGOCL	=1 is flag to begin on-call surf wave (subroutine EXEC)
BEGNSL	=1 is flag to begin non-sched surf wave (subroutine EXEC)
BEGSH	=1 is flag to begin sched helo wave (subroutine EXEC)
BEGSL	=1 is flag to begin sched surf wave (subroutine EXEC)
BSPD(I)	Boat speed (I=type boat)
C	Class of wave(1=sched, 2=on-call, 3=non sched)
CA	Total # of lines of serial data for all on-call helo waves
CC	Preserves C
CC1	Class of wave in subroutine HCKLD
CHOUR	Time for starting first LCAC wave
CLZDIS	Distance from shoreline to LCAC LZ (nm)
CS	Total # of lines of serial data for all on-call surf waves
CSPD	LCAC speed over ground
DBAT	Tally # of boats into landing zone
DEST	Destination
DLABAT	Delay table
DLADIS	Distance boat must travel from ship to shore(nm)
DLTIME	Time needed to load wave (subroutine HRLD)
DS	Deck spots available code (0=no, 1=yes)

DTIME	Time interval to unload LZ
DTM(I)	Value assigned from LTF(I,J) (boat time factors)
DT	Time interval for wave to return to ship
EASH	Running total of the type load
FCH	# of first line of serials to be carried in on-call helo wave
FCL	# of first line of serials to be carried in on-call surf wave
FMH	# of first line of serials to be carried in non-sched helo wave
FML	# of first line of serials to be carried in non-sched surf wave
FSH	# of first line of serials to be carried in sched helo wave
FSL	# of first line of serials to be carried in sched surf wave
H1CLK(W,N)	Sched subarray for helo info (w=wave #, n=wave info -see separate explanation page)
H2CLK(W,N)	On-call subarray for helo info (w=wave #, n=wave info -see separate explanation page)
H3CLK(W,N)	Non-sched subarray for helo info (w=wave #,n=wave info -see separate explanation page)
HCWAVE	Total # on-call helo waves
HEST(LDT,TYP)	Temp array for same info as HOST(LDT,TYP)
HGF(I)	# helos by type that can land and unload simultaneously in LZ (I=helo type)
HHOUR	Time for starting first helo wave
HI	Very large value to prevent info to be considered in time sort (32000)
HNWAVE	Total # non-sched helo waves
HOLD	Same as HI (=16000)
HOS(I,J,K)	Operational deck spots for helo ops (I=ship type, J=type loading: internal,external, comb; K=type helo)
HOST(LDT,TYP)	Total # of all of deck spots (LDT=type loading:internal, external, combination; TYP=type helo)
HOUR	Current time in minutes
HSPD(I)	Helo speeds (I=type load)
HSWAVE	Total # sched helo waves

HTF(I)	Table of helo time factors (see separate table)
I	Counter
LATE	Very large value to prevent info to be considered in time sort (17000)
L1CLK(W,N)	Sched subarray for surf wave info (W=wave#, N=wave data-see separate explanation)
L2CLK(W,N)	On-call subarray for surf wave info (W=wave #, N=wave data -see separate explanation
L3CLK(W,N)	Non-sched subarray for surf wave info (W=wave #, N=wave data- see separate explanation
LCWAVE	Total # of on-call boat waves
LDT	Wave load factor (1=internal, 2=external, 3=combination)
LGF(I)	# of boats by type that can unload simultaneously at numbered beach or LCAC LZ
LHOUR	Time for starting first boat wave (excluding LCACs)
LNWAVE	Total # of non-sched boat waves
LS	Wave # = SSL(N,2)
LSWAVE	Total # of sched boat waves
LTF(I)	Boat time factors (I=boat wave operation, J=type boat)
LZDIS	Distance from shoreline to helo LZ (nm)
M	1st line of serial data to be carried by wave (M=FSH or FCH or FMH)
MCLOCK(I,N)	Master clock subarray (i=1 to 6, 1=sched helo, 2=on-call helo, 3=non-sched helo, 4=sched surf, 5=on-call surf, 6= non-sched surf; n=wave data-see separate explanation)
MODE	Air=1, surf=2
MS	Total # of lines of serial data for non-sched surf waves
NA	Total # of lines of serial data for non-sched helo waves
NBT	# boat types (5)
NCAR(I)	# carriers of type I
NCLZ	# of LCAC LZ's available to current landing plan
NCT	# LCAC types (1)
NH	# of helos returning by type
NHT	# of helo types(4)
NLSET	Flag to signal begin non-sched surf waves after on-call waves completed

NLT	# of LVT types (1)
NLZ	# of LZ's available to current landing plan
NNB	# of numbered beaches available to current landing plan
NS	# of deck spots to be used for this wave (subroutine HRLD)
NSA	Counter for non-sched helo input
NSS	Counter for non-sched surf input
NSSET	Flag to signal begin non-sched helo waves after on-call waves completed
NST	# ships total from which can off load
NUS(I)	# of ships by type from which off-loading occurs (I=type of ship, 1=lha, 2=lph, 3=lsd, 4=lpd, 5=lst)
OA	counters for on-call helo serials
OCHCK	Signal identifying whether on-call helos are to follow scheduled (OCHCK =0) or break into sched wave sequence (OCHCK=2).If =2 then times must be specified for each on-call wave to commence operation
OCHTAB(I)	Time for on-call helo wave to commence operation if not following sequence (I=wave #). Scheduled waves are delayed until on-call operations completed
OCLCK	Signal identifying whether on-call boats are to follow scheduled (OCLCK =0) or break into sched wave sequence (OCLCK=2). If =2 then times must be specified for each on-call wave to commence operation
OCSET	Signal
OCTIM	L1CLK(N,2) + ZTIME = time next event plus time to unload on beaches
OS	Counters for on-call surf serials
PAC	# and type(load) of deck spots in use by the current wave prior to launch
PASH	Running total of personnel ashore
P(I)	Pools of helos (i=helo type)
RMFRC	Fractional availability of deck spots
SA	Counters for sched helo serialized data
SCA(I,J)	Storage arrays for serialized input data for on-call helo waves. (J=following values:0=WUV, 1=wave #, 2=personnel, 3=type carrier, 4=# of carriers, 5=type load, 6=destination, 7=launch signal; I=# of serial line of data)
SCH(I,1)	Wave # of each line of data for on-call helo wave (I=wave#)
	waves completed

SCH(I,2)	First line of serialized data for each on-call helo wave (I=wave #)
SCL(I,1)	Type carrier for on-call surf wave (I=wave #)
SCL(I,2)	First line of serialized data for each on-call surf wave (I=wave #)
SCS(I,J)	Storage arrays for serialized input data for on-call surf waves. (J=following values:0=WUV, 1=wave #, 2=personnel, 3=type carrier, 4=# of carriers, 5=type load, 6=destination, 7=launch signal; I=# of serial line of data)
SEASH	Total equipment ashore
SFASH	Running total of firepower ashore
SNA(I,J)	Storage arrays for serialized input data for non-sched helo waves. (J=following values:0=WUV, 1=wave #, 2=personnel, 3=type carrier, 4=# of carriers, 5=type load, 6=destination, 7=launch signal; I=# of serial line of data)
SNH(I,1)	Wave # of each line of data for non-sched helo wave (I=wave #)
SNH(I,2)	First line of serialized data for each non-sched helo wave (I=wave #)
SSH(I,1)	Wave # of each line of data for sched helo wave (I=wave #)
SSH(I,2)	First line of serialized data for each sched helo wave (I=wave #)
SNL(I,1)	Type carrier for non-sched surf wave (I=wave #)
SNL(I,2)	First line of serialized data for each non-sched surf wave (I=wave #)
SNS(I,J)	Storage arrays for serialized input data for non-sched surf waves. (J=following values:0=WUV, 1=wave #, 2=personnel, 3=type carrier, 4=# of carriers, 5=type load, 6=destination, 7=launch signal; I=# of serial line of data)
SS	Counters for sched surf serialized data
SSA(I,J)	Storage arrays for serialized input data for sched helo waves. (J=following values:0=WUV, 1=wave #, 2=personnel, 3=type carrier, 4=# of carriers, 5=type load, 6=destination, 7=launch signal; I=# of serial line of data)
SSET	Flag for calling restart
SSL(I,1)	Type carrier for sched surf wave (I=wave #)
SSL(I,2)	First line of serialized data for each sched surf wave (I=wave #)
SSPD	Ship speed

SSS(I,J)	Storage arrays for serialized input data for sched surf waves. (J=following values:0=WUV, 1=wave #, 2=personnel, 3=type carrier, 4=# of carriers, 5=type load, 6=destination, 7=launch signal; I=# of serial line of data)
TA	Total # lines of serialized data for sched helo waves
TRNH1	The # of the first scheduled helo wave in first turnaround wave
TRNS1	The # of the first scheduled surf wave in first turnaround wave
TS	Total # lines of serialized data for sched surf waves
VLADIS	Distance from launch to shore of landing craft
W	Counter for wave #
WD	Well deck availability code (0=no, 1=yes)
WDS(I)	Well deck spot array (I=type ship)
WDST	Well deck spots total in the fleet
WEST	Well deck spots total in the fleet
WLDT	Load type being employed by current wave
WNS	# deck spots being employed by current wave
WSET	Signal to call restart (=1)
WTYP	Type carrier being employed by current wave
WW	Preserve w (wave #) in subroutine HRLD
WW1	Signal used to setup (subroutine HCKLD)
ZN(I)	Time table for landing in zone (I) (time unload complete)
ZONE(I)	Landing zone signal (I=LZ type) (1=free to land in)
ZTIME	Max time to unload over all landing zones in use

H. WAVE CLOCK PARAMETERS

The wave clock parameters are found in six arrays. There is an array for each class (3) and for both helo and surface modes. They are of the form H1CLK(W,N). The table below is a breakdown of the coding for N.

N Value	Value and code for wave clock parameter
0	Signal: 0=wave not loaded yet 1=wave is or has loaded 2=wave has offloaded cargo ashore
1	Wave number
2	Time of next event (future time)
3	Code for next event: 1=start loading wave 2=launch loaded wave 3=reach destination 4=complete unload and start return ship 5=arrive back at ship area
4	Wave load factor (helo only): 1=internal loading 2=external loading 3=combination load
5	Mode of transportaion (use by master clock): 1=helo 2=surface
6	Class of serial (master clock): 1=scheduled 2=on-call 3=non-scheduled
6+I	Number of helos of type I in wave

END OF FILE

APPENDIX F

PROGRAM LISTING

```
5 'PROGRAM THAT WRITES TO FILE ;27 APR 84:DATSHIP.BAS
6 'SM RITACCO
7 'INITIAL INPUT ROUTINE LINES 0-470
20 DEF FNX$=MID$(A1$,INSTR(A1$,",")+1)
30 DEF FNY$=STR$(L)+CHR$(4)+A1$
40 DEFINT A-Y:INPUT"# OF SERIAL LINES OF DATA FOR SCHED,ON-CALL, % NONSCHED HELO
 THEN SURFACE:",TA,CA,NA,TS,CS,MS
42 DIM SCA(CA,7),SSA(TA,7),SSS(TS,7),SCS(CS,7),SNA(NA,7),SNS(MS,7),HOS(5,3,4)
44 DIM NUS(5),HTF(8,3),LTF(8,6),HSPD(3),BSDP(6),DLABAT(4,3,7),HOST(3,4)
50 PRINT"Setup";FRE(9):OPEN"I",1,"T-1.DAT
60 FOR I=1 TO 5:FOR J=1 TO 3:FOR K=1 TO 4:INPUT#1,HOS(I,J,K)
62 NEXT:NEXT:NEXT:INPUT"# OF SHIPS BY TYPE (LHA,LPH,LSD,LPD,LST):",NUS(1),NUS(2)
,NUS(3),NUS(4),NUS(5)
70 FOR I=1 TO 8:FOR J=1 TO 3:INPUT#1, HTF(I,J):NEXT:NEXT
80 NHT=4:NLT=1:NBT=5:NCT=0:NST=5:TRNS1=10:TRNS2=10:INPUT"1ST TURNAROUND SCHED HE
LO WAVE:",TRNH1
82 INPUT"# OF BEACHES AND # OF LANDING ZONES:",NNB,NLZ
90 FOR J=1 TO 3:INPUT#1, HSPD(J):NEXT
100 FOR I=1 TO NLT+NBT+NCT:INPUT#1,BSPD(I):NEXTRUN
101 INPUT"HELO LAUNCH DISTANCE, BOAT LAUNCH DISTANCE, AND SHORE TO LZ DISTANCE:"
,ALADIS,LODDIS,LZDIS
102 NEXT:SSPD=20:ASADIS=8:VLADIS=2:DLADIS=8:CLZDIS=1
104 CSPD=10
106 INPUT"LHOUR AND HHOUR:",LHOUR,HHOUR
110 INPUT"# OF WAVES IN ORDER SCHED, ON-CALL, NONSCHED HELO THEN SURFACE:",HSWAV
E,HCWAVE,HNWAVE,LSWAVE,LCWAVE,LNWAVE
112 CHOUR=17000:LATE=17000:HI=32000:HOLD=16000:VS=1
114 VC=2:VN=3:DIM SSH(HSWAVE,2),SCH(HCWAVE,2),SNH(HNWAVE,2),SSL(LSWAVE,2)
120 DIM SCL(LCWAVE,2),SNL(LNWAVE,2)
130 DIM H1CLK(HSWAVE,NHT+6),H2CLK(HCWAVE,NHT+6),H3CLK(HNWAVE,NHT+6)
132 DIM L1CLK(LSWAVE,NLT+NBT+NCT+6),L2CLK(LCWAVE,NLT+NBT+NCT+6)
134 DIM L3CLK(LNWAVE,NLT+NBT+NCT+6),OCHTAB(HCWAVE)
140 DIM AT(NHT+NLT+NBT+NCT),AP(NHT+NLT+NBT+NCT),P(NHT+NLT+NBT+NCT),HGF(NHT)
142 DIM LGF(NLT+NBT+NCT),WDS(NST):INPUT"# OF HELOS THAT CAN OCCUPY AN LZ AT ANY
ONE TIME BY TYPE (CH46, CH53D, CH53E, UH1N):",HGF(1),HGF(2),HGF(3),HGF(4)
144 WDS(1)=6:WDS(2)=2:WDS(3)=3:WDS(4)=5:WDS(5)=7:OCHTAB(1)=18
150 OCHCK=0
180 INPUT"# OF CARRIERS BY TYPE (CH46,CH53D,CH53E,UH1N,LVT,LCM6,LCM8,LCU,LST,LAR
C):",AT(1),AT(2),AT(3),AT(4),AT(5),AT(6),AT(7),AT(8),AT(9),AT(10)
190 INPUT"# OF SURFACE CRAFT THAT CAN OCCUPY A BEACH AT ANY ONE TIME BY TYPE (LV
T,LCM6,LCM8,LCU,LST,LARC):",LGF(1),LGF(2),LGF(3),LGF(4),LGF(5),LGF(6)
220 FOR I=1 TO 8:FOR J=1 TO 6:INPUT#1, LTF(I,J):NEXT:NEXT:CLOSE 1
230 PRINT CHR$(12);
232 'IF EOF(2) THEN END
240 ZZ$="SETUP-A"
250 A$="LFE"
252 INPUT"INPUT DATA FILE NAME:",INFILE$
260 OPEN"I",1,INFILE$
262 INPUT "OUTPUT FILE NAME:",OUTFILE$
270 SS=1:SA=1:OS=1:OA=1:NSS=1:NSA=1
280 INPUT#1,A$:LINE INPUT#1,A1$:IF LEFT$(A$,1)="E" THEN CLOSE 1:NSA=NSA-1:GOTO 4
70
290 IF LEFT$(A$,1)="T"THEN 280
300 FOR I=1 TO 5:Z(I)=VAL(A1$):A1$=MID$(A1$,INSTR(A1$,",")+1):NEXT:Z(0)=Z(0)
302 IF LEFT$(A1$,1)="S"THEN GOSUB 320 ELSE GOSUB 360
304 TWUV!=TWUV!+WUV
310 GOTO 280
320 IF RIGHT$(A$,1)="O"THEN 340 ELSE IF RIGHT$(A$,1)="N"THEN 350
330 A1$=MID$(A1$,3):FOR J=0 TO 5
332 Z(5)=0:Z(4)=0:SSS(SS,J)=Z(J):NEXT:SSS(SS,1)=VAL(A$):K=3
```

```
334 FOR J=1 TO 8 STEP 2:X$=FNX$
336 GOSUB 400:SSS(SS,K)=L:K=K+1:NEXT:SSS(SS,0)=WUV:SS=SS+1:RETURN
340 A1$=MID$(A1$,3):FOR J=0 TO 5
342 Z(5)=0:Z(4)=0:SCS(OS,J)=Z(J):NEXT:SCS(OS,1)=VAL(A$):K=3
344 FOR J=1 TO 8 STEP 2:X$=FNX$
346 GOSUB 400:SCS(OS,K)=L:K=K+1:NEXT:SCS(OS,0)=WUV:OS=OS+1:RETURN
350 A1$=MID$(A1$,3):FOR J=0 TO 5
352 Z(5)=0:Z(4)=0:SNS(NSS,J)=Z(J):NEXT:SNS(NSS,1)=VAL(A$):K=3
354 FOR J=1 TO 8 STEP 2:X$=FNX$
356 GOSUB 400:SNS(NSS,K)=L:K=K+1:NEXT:SNS(NSS,0)=WUV:NSS=NSS+1:RETURN
360 IF RIGHT$(A$,1)="O"THEN 380 ELSE IF RIGHT$(A$,1)="N"THEN 390
370 A1$=MID$(A1$,3):FOR J=0 TO 5
372 Z(5)=0:Z(4)=0:SSA(SA,J)=Z(J):NEXT:SSA(SA,1)=VAL(A$):K=3
374 FOR J=1 TO 8 STEP 2:X$=FNX$
376 GOSUB 400:SSA(SA,K)=L:K=K+1:NEXT:SSA(SA,0)=WUV:SA=SA+1:RETURN
380 A1$=MID$(A1$,3):FOR J=0 TO 5
382 Z(5)=0:Z(4)=0:SCA(OA,J)=Z(J):NEXT:SCA(OA,1)=VAL(A$):K=3
384 FOR J=1 TO 8 STEP 2:X$=FNX$
386 GOSUB 400:SCA(OA,K)=L:K=K+1:NEXT:SCA(OA,0)=WUV:OA=OA+1:RETURN
390 A1$=MID$(A1$,3):FOR J=0 TO 5
392 Z(5)=0:Z(4)=0:SNA(NSA,J)=Z(J):NEXT:SNA(NSA,1)=VAL(A$):K=3
394 FOR J=1 TO 8 STEP 2:X$=FNX$
396 GOSUB 400:SNA(NSA,K)=L:K=K+1:NEXT:SNA(NSA,0)=WUV:NSA=NSA+1:RETURN
400 IF K=4 THEN L=VAL(A1$):A1$=FNX$:RETURN
410 IF K=3 THEN L=ASC(LEFT$(A1$,1))-64:A1$=FNX$:RETURN
420 IF K<>6 THEN 440 ELSE GOSUB 460:X$=RIGHT$(A1$,3):IF X$="B-1" OR X$="R-1" THE
N L=1 ELSE IF X$="B-2" OR X$="R-2" THEN L=2 ELSE IF X$="B-3" OR X$="R-3" THEN L=
3 ELSE L=4
430 GOTO 450
440 X$=LEFT$(A1$,1):IF X$="C"THEN L=3 ELSE IF X$="I"THEN L=1 ELSE IF X$="E" THEN
 L=2
450 A1$=FNX$:RETURN
460 WUV$=FNX$:A1$=LEFT$(A1$,LEN(A1$)-LEN(WUV$)-1):WUV=VAL(WUV$):RETURN
469 'INITIALIZES TIME CLOCKS AND TIME CLOCK PARAMETERS LINES 470-1280
470 OPEN OUTFILE$ FOR OUTPUT AS #2
480 LLHOUR=LHOUR:HHHOUR=HHOUR
490 T(1)=LLHOUR:T(2)=HHHOUR:T(3)=CCHOUR:FOR I=1 TO 3:U(I)=4-I:NEXT:FOR J=1 TO 2:
FOR I=1 TO 3:IF T(I+1)>T(I) THEN SWAP T(I),T(I+1):SWAP U(I),U(I+1)
500 NEXT:NEXT:FOR I=1 TO 3:T(I)=T(I)-T(3):ON U(I) GOSUB 510,520,530:NEXT
502 ERASE T:ERASE U:GOTO 540
510 CCHOUR=T(I):RETURN
520 HHHOUR=T(I):RETURN
530 LLHOUR=T(I):RETURN
540 GOSUB 550:GOTO 2270
550 'AT(1)=180:AT(2)=100:AT(3)=100:AT(4)=50:AT(5)=200:AT(6)=81:AT(7)=64:TRNS1=10
552 'TRNH1=7:N=NHT+NLT+NBT+NCT:FOR I=1 TO N:AP(I)=AT(I):NEXT
560 N=NHT+NLT+NBT+NCT:FOR I=1 TO N:AP(I)=AT(I):NEXT
580 FOR I=1 TO 6:MCLOCK(I,2)=LATE:NEXT
590 FOR I=1 TO HSWAVE:H1CLK(I,2)=LATE:NEXT:FOR I=1 TO HCWAVE
592 H2CLK(I,2)=LATE:NEXT:FOR I=1 TO HNWAVE:H3CLK(I,2)=LATE:NEXT
594 FOR I=1 TO LSWAVE:L1CLK(I,2)=LATE:NEXT:FOR I=1 TO LCWAVE:L2CLK(I,2)=LATE
600 NEXT:FOR I=1 TO LNWAVE:L3CLK(I,2)=LATE:NEXT
610 FOR K=1 TO NLZ:ZN(K)=0:NEXT
620 GOSUB 1630
630 J=HHHOUR
640 FOR N=1 TO TRNH1-1
650 H1CLK(N,1)=N:H1CLK(N,2)=J:H1CLK(N,3)=1:H1CLK(N,5)=1:H1CLK(N,6)=1:J=J+1
660 NEXT:GOSUB 1290
670 IF OCHCK<2 THEN 710
680 FOR I=1 TO HCWAVE
```

```
690 H2CLK(I,1)=I:H2CLK(I,2)=OCHTAB(I)+HHHOUR:H2CLK(I,3)=1:H2CLK(I,5)=1
692 H2CLK(I,6)=2
700 NEXT:FCH=1:GOSUB 1360:GOTO 720
710 GOTO 720
720 L=NLT+NBT+NCT:L1CLK(1,2)=LLHOUR
730 FOR N=1 TO LSWAVE
740 L1CLK(N,1)=N:L1CLK(N,3)=2:L1CLK(N,5)=2:L1CLK(N,6)=1
750 LS=SSL(N,2):FOR I=1 TO L:NCAR(I)=0:P(I+NHT)=AP(I+NHT):FOR K=1 TO NNB
752 DBAT(I,K)=0:NEXT:NEXT
760 IF LS=TS THEN 820 ELSE IF SSS(LS,1)<>N THEN 820 ELSE I=SSS(LS,3)-NHT
770 IF I<=0 THEN PRINT"NO MO DATA":STOP
780 IF P(I+NHT)>=SSS(LS,4)THEN 810 ELSE J=SSS(LS,1):L1CLK(J,2)=HOLD
790 IF SSS(LS-1,1)<>SSS(LS,1)THEN 800 ELSE LS=LS-1:SSS(LS,7)=0:GOTO 790
800 FSL=LS:PRINT"LS";LS;" - Insufficient Scheduled landing craft":STOP
810 NCAR(I)=NCAR(I)+SSS(LS,4):SSS(LS,7)=1:DEST=SSS(LS,6)
812 DBAT(I,DEST)=DBAT(I,DEST)+SSS(LS,4):P(I+NHT)=P(I+NHT)-SSS(LS.4):LS=LS+1
814 GOTO 760
820 FOR I=1 TO L:AP(I+NHT)=P(I+NHT):L1CLK(N,I+6)=NCAR(I):IF NCAR(I)<>0 THEN II=I
830 NEXT
840 FOR K=1 TO NNB:DTIME(K)=0:FOR I=1 TO L
850 IF DBAT(I,K)<=0 THEN 880
860 DBAT(I,K)=DBAT(I,K)-LGF(I)
870 DTIME(K)=DTIME(K)+LTF(1,I)+LTF(2,I)+LTF(8,I):GOTO 850
880 NEXT:NEXT
890 ZTIME=0:FOR K=1 TO L:IF DTIME(K)>ZTIME THEN ZTIME=DTIME(K)
900 IF N=LSWAVE THEN OCTIM=L1CLK(N,2)+ZTIME:GOTO 950
910 NEXT:IF N=LSWAVE THEN OCTIM=L1CLK(N,2)+ZTIME:GOTO 950
920 L1CLK(N+1,2)=L1CLK(N,2)+ZTIME
930 IF SSS(LS,2)<>0 THEN 940 ELSE L1CLK(N+1,2)=HOLD:GOTO 950
940 NEXT
950 IF NCT=0 THEN PRINT"Boat pool";AP(5);AP(6);AP(7);AP(8);AP(9):AP(10) ELSE PRI
NT "Boat pool";AP(5);AP(6);AP(7);AP(8);AP(9);AP(10);AP(11)
952 GOSUB 1520
956 IF CS=0 THEN FCL=HI:FML=HI:GOTO 1226
960 IF SCS(1,1)=0 THEN FCL=HI:FML=HI:GOTO 1226
970 L=NLT+NBT+NCT:L2CLK(1,2)=OCTIM
980 FOR N=1 TO LCWAVE
990 L2CLK(N,1)=1:L2CLK(N,3)=2:L2CLK(N,5)=2:L2CLK(N,6)=2
1000 LC=SCL(N,2):FOR I=1 TO L:NCAR(I)=0:P(I+NHT)=AP(I+NHT):FOR K=1 TO NNB
1002 DBAT(I,K)=0:NEXT:NEXT
1010 IF LC=CS THEN 1100 ELSE IF SCS(LC,1)<>N THEN 1100
1020 I=SCS(LC,3)-NHT:IF I<=0 THEN LPRINT"NO DATA":GOTO 1226
1030 IF P(I+NHT)>=SCS(LC,4)THEN 1090
1040 J=SCS(LC,1):L2CLK(J,2)=HOLD
1050 IF SCS(LC-1,1)<>SCS(LC,1)THEN 1070
1060 LC=LC-1:SCS(LC,7)=0:GOTO 1050
1070 FCL=LC:LPRINT"LC";LC;" Insufficient on-call craft"
1080 GOTO 1230
1090 NCAR(I)=NCAR(I)+SCS(LC,4):P(I+NHT)=P(I+NHT)-SCS(LC,4):SCS(LC,7)=1
1092 DEST=SCS(LC,6):DBAT(I,DEST)=DBAT(I,DEST)+SCS(LC,4):LC=LC+1:GOTO 1010
1100 FOR I=1 TO L:AP(I+NHT)=P(I+NHT):L2CLK(N,I+6)=NCAR(I)
1102 IF NCAR(I)<>0 THEN II=I
1110 NEXT
1120 FOR K=1 TO NNB:DTIME(K)=0:FOR I=1 TO L
1130 IF DBAT(I,K)<=0 THEN 1160
1140 DBAT(I,K)=DBAT(I,K)-LGF(I)
1150 DTIME(K)=DTIME(K)+LTF(1,I)+LTF(2,I)+LTF(8,I):GOTO 1130
1160 NEXT:NEXT
1170 ZTIME=0:FOR K=1 TO L:IF DTIME(K)>ZTIME THEN ZTIME=DTIME(K)
1180 NEXT
```

```
1182 IF N=LCWAVE THEN NSTIM=L2CLK(N,2)+ZTIME:GOTO 1220
1190 L2CLK(N+1,2)=L2CLK(N,2)+ZTIME
1200 IF SCS(LC,2)<>0 THEN 1210 ELSE L2CLK(N+1,2)=HOLD:GOTO 1220
1210 NEXT
1220 IF NCT=0 THEN PRINT
1222 GOSUB 1580
1226 IF MS=0 THEN FML=HI:GOTO 1240
1230 IF SNS(1,1)=0 THEN FML=HI:L3CLK(1,2)=LATE
1240 FOR I=1 TO NST:WDST=WDST+WDS(I)*NUS(I):NEXT:WEST=WDST
1250 PASH=0:SEASH=0:SFASH=0:FOR I=1 TO NST:FOR J=1 TO 3:FOR K=1 TO NHT
1252 HOST(J,K)=HOST(J,K)+HOS(I,J,K)*NUS(I):HEST(J,K)=HOST(J,K)
1254 NEXT:NEXT:NEXT:FOR I=1 TO 6:IF I<=3 THEN MCLOCK(I,5)=1 ELSE MCLOCK(I,5)=2
1260 IF I=1 OR I=4 THEN MCLOCK(I,6)=1 ELSE IF I=2 OR I=5 THEN MCLOCK(I,6)=2 ELSE
 MCLOCK(I,6)=3
1270 NEXT
1272 DS=1
1280 RETURN
1289 '
1290 TIME=LATE:FOR W=1 TO HSWAVE
1300 IF H1CLK(W,2)>=TIME THEN 1320
1310 TIME=H1CLK(W,2):ACODE=H1CLK(W,3):HWAVE=W
1320 NEXT
1330 MCLOCK(1,2)=TIME:MCLOCK(1,1)=HWAVE:MCLOCK(1,3)=ACODE:H1CLK(HWAVE,2)=HOLD
1332 MCLOCK(1,5)=1:MCLOCK(1,6)=1
1340 IF ACODE=6 THEN H1CLK(HWAVE,2)=LATE
1350 RETURN
1359 '
1360 'PRINT"HCSORT"
1370 TIME=LATE:FOR W=1 TO HCWAVE
1380 IF H2CLK(W,2)>=TIME THEN 1400
1390 TIME=H2CLK(W,2):ACODE=H2CLK(W,3):HWAVE=W
1400 NEXT
1410 MCLOCK(2,2)=TIME:MCLOCK(2,1)=HWAVE:MCLOCK(2,3)=ACODE:H2CLK(HWAVE,2)=HOLD
1412 MCLOCK(2,5)=1:MCLOCK(2,6)=2
1420 IF ACODE=6 THEN H2CLK(HWAVE,2)=HOLD
1430 RETURN
1439 '
1440 'PRINT"HNSORT"
1450 TIME=LATE:FOR W=1 TO HNWAVE
1460 IF H3CLK(W,2)>=TIME THEN 1480
1470 TIME=H3CLK(W,2):ACODE=H3CLK(W,3):HWAVE=W
1480 NEXT
1490 MCLOCK(3,2)=TIME:MCLOCK(3,1)=HWAVE:MCLOCK(3,3)=ACODE:H3CLK(HWAVE,2)=HOLD
1492 MCLOCK(3,5)=1:MCLOCK(3,6)=3
1500 IF ACODE=6 THEN H3CLK(HWAVE,2)=LATE
1510 RETURN
1519 '
1520 'PRINT"LSSORT"
1530 TIME=LATE:FOR W=1 TO LSWAVE:IF L1CLK(W,2)>=TIME THEN 1550
1540 TIME=L1CLK(W,2):ACODE=L1CLK(W,3):LWAVE=W
1550 NEXT
1560 MCLOCK(4,2)=TIME:MCLOCK(4,1)=LWAVE:MCLOCK(4,3)=ACODE:L1CLK(LWAVE,2)=HOLD
1562 MCLOCK(4,5)=2:MCLOCK(4,6)=1:IF ACODE=6 THEN L1CLK(WAVE,2)=LATE
1570 RETURN
1579 '
1580 'PRINT"LCSORT"
1590 TIME=LATE:FOR W=1 TO LCWAVE:IF L2CLK(W,2)>=TIME THEN 1610
1600 TIME=L2CLK(W,2):ACODE=L2CLK(W,3):LWAVE=W
1610 NEXT:MCLOCK(5,2)=TIME:MCLOCK(5,1)=LWAVE:MCLOCK(5,3)=ACODE:L2CLK(LWAVE,2)=HO
LD
```

```
1612 MCLOCK(5,5)=2:MCLOCK(5,6)=2:IF ACODE=6 THEN L2CLK(LWAVE,2)=LATE
1620 RETURN
1628 '
1629 'COMPUTES ARRAYS FOR WAVE #'S AND SERIAL # OF 1ST SERIAL IN WAVE
1630 IF TA=0 THEN 1680 ELSE W=1:SSH(W,2)=1:SSH(W,1)=W:IF TA=0 THEN 1680
1640 FOR I=1 TO TA:IF SSA(I,1)=W THEN 1670
1650 IF SSA(I,1)=0 THEN 1680
1660 W=W+1:SSH(W,2)=I:SSH(W,1)=W
1670 NEXT
1680 IF CA=0 THEN 1730 ELSE W=1:SCH(W,2)=1:SCH(W,1)=W:IF CA=0 THEN 1730
1690 FOR I=1 TO CA:IF SCA(I,1)=W THEN 1720
1700 IF SCA(I,1)=0 THEN 1730
1710 W=W+1:SCH(W,2)=I:SCH(W,1)=W
1720 NEXT
1730 IF NA=0 THEN 1780 ELSE W=1:SNH(W,2)=1:SNH(W,1)=W:IF NA=0 THEN 1780
1740 FOR I=1 TO NA:IF SNA(I,1)=W THEN 1770
1750 IF SNA(I,1)=0 THEN 1780
1760 W=W+1:SNH(W,2)=I:SNH(W,1)=W
1770 NEXT
1780 IF TS=0 THEN 1830 ELSE W=1:SSL(W,2)=1:SSL(W,1)=SSS(1,3)
1790 FOR I=1 TO TS:IF SSS(I,1)=W THEN 1820
1800 IF SSS(I,1)=0 THEN 1830
1810 W=W+1:SSL(W,2)=I:SSL(W,1)=SSS(I,3)
1820 NEXT
1830 IF CS=0 THEN 1880 ELSE W=1:SCL(W,2)=1:SCL(W,1)=SCS(1,3)
1840 FOR I=1 TO CS:IF SCS(I,1)=W THEN 1870
1850 IF SCS(I,1)=0 THEN 1880
1860 W=W+1:SCL(W,2)=I:SCL(W,1)=SCS(I,3)
1870 NEXT
1880 IF MS=0 THEN 1930 ELSE W=1:SNL(W,2)=1:SNL(W,1)=SNS(1,3)
1890 FOR I=1 TO MS:IF SNS(I,1)=W THEN 1920
1900 IF SNS(I,1)=0 THEN 1930
1910 W=W+1:SNL(W,2)=I:SNL(W,1)=SNS(I,3)
1920 NEXT
1930 RETURN
1939 '
1940 'EXEC SUBROUTINE-CONTROLS TIMING AND EXECUTION OF MODEL
2270 IF HOUR=LATE THEN CLOSE #2:STOP ELSE PTIME=32000
2280 IF PTIME<HOUR THEN CLOSE #2:STOP
2290 IF CC1<>0 THEN GOSUB 8390
2300 IF BEGNSH=1 THEN GOSUB 7820
2310 IF BEGOCH=1 THEN GOSUB 7750
2320 IF BEGSH=1 THEN GOSUB 7860
2330 IF WSET<>0 THEN GOSUB 7080
2340 GOSUB 3630:IF MODE=2 THEN 2450
2350 IF A>1 THEN 2370
2360 GOSUB 2550:GOTO 2270
2370 IF A>2 THEN 2390
2380 GOSUB 3700:GOTO 2270
2390 IF A>3 THEN 2410
2400 GOSUB 4060:GOTO 2270
2410 IF A>4 THEN 2430
2420 GOSUB 4610:GOTO 2270
2430 IF A>5 THEN STOP
2440 GOSUB 4270:GOTO 2270
2450 IF A>1 THEN 2470
2460 GOSUB 4990:GOTO 2270
2470 IF A>2 THEN 2490
2480 GOSUB 5400:GOTO 2270
2490 IF A>3 THEN 2510
```

```
2500 GOSUB 5790:GOTO 2270
2510 IF A<>4 THEN 2530
2520 GOSUB 6160:GOTO 2270
2530 IF A<>5 THEN STOP
2540 GOSUB 6630:GOTO 2270
2549 '
2550 'HRLD
2560 FOR I=1 TO NHT:NCAR(I)=0:P(I)=AP(I):NEXT:LDT=0
2570 IF C>1 THEN 2780
2580 M=SSH(W,2):W=SSA(M,1)
2590 IF DS<>0 THEN 2610 ELSE H1CLK(W,3)=1:H1CLK(W,2)=HOUR+10:FSH=SSH(W,2)
2600 GOSUB 3400:RETURN
2610 IF SSA(M,1)<>W THEN 2680
2620 I=SSA(M,3)
2630 IF P(I)>=SSA(M,4)THEN 2650
2640 H1CLK(W,1)=W:H1CLK(W,2)=HOUR+10:H1CLK(W,3)=1:GOSUB 3400:RETURN
2650 P(I)=P(I)-SSA(M,4):NCAR(I)=NCAR(I)+SSA(M,4)
2660 IF SSA(M,5)>LDT THEN LDT=SSA(M,5)
2670 IF M=TA THEN 2680 ELSE M=M+1:GOTO 2610
2680 H1CLK(W,1)=W:H1CLK(W,3)=2:H1CLK(W,4)=LDT:FOR I=1 TO NHT
2682 AP(I)=P(I):H1CLK(W,I+6)=NCAR(I):NEXT
2690 FOR I=1 TO NHT:IF NCAR(I)<>0 THEN II=I
2700 NEXT
2710 H1CLK(W,0)=1
2720 GOSUB 3200
2730 IF NS=0 THEN DLTIME=10:H1CLK(W,3)=1:H1CLK(W,0)=0
2740 IF DS=0 THEN CC=1:WW=W:WSET=0:WTYP=II:WNS=NS:WLDT=LDT
2750 H1CLK(W,2)=HOUR+DLTIME:FSH=M:IF W=HSWAVE AND OCHCK=0 THEN OCSET=1:BEGOCH=1
ELSE IF W=HSWAVE THEN NSSET=1:BEGNSH=1
2760 GOSUB 3400
2770 RETURN
2780 IF C>2 THEN 3010
2790 MC=SCH(W,2):W=SCA(MC,1)
2800 IF OCHCK=2 AND SSET=0 THEN GOSUB 7900
2810 IF DS<>0 THEN 2830 ELSE H2CLK(W,3)=1:H2CLK(W,2)=HOUR+10:FCH=SCH(W,2)
2820 GOSUB 3480:RETURN
2830 IF SCA(MC,1)<>W THEN 2900
2840 I=SCA(MC,3)
2850 IF P(I)>=SCA(MC,4) THEN 2870
2860 H2CLK(W,1)=W:H2CLK(W,2)=HOUR+10:H2CLK(W,3)=1:GOSUB 3480:RETURN
2870 P(I)=P(I)-SCA(MC,4):NCAR(I)=NCAR(I)+SCA(MC,4)
2880 IF SCA(MC,5)>LDT THEN LDT=SCA(MC,5)
2890 IF MC=CA THEN 2900 ELSE MC=MC+1:GOTO 2830
2900 H2CLK(W,1)=W:H2CLK(W,3)=2:H2CLK(W,4)=LDT:FOR I=1 TO NHT
2902 AP(I)=P(I):H2CLK(W,I+6)=NCAR(I):NEXT
2910 FOR I=1 TO NHT:IF NCAR(I)<>0 THEN II=I
2920 NEXT
2930 H2CLK(W,0)=1
2940 IF W=HCWAVE AND OCHCK=2 AND SSET=1 THEN SSET=2:BEGSH=1
2950 GOSUB 3200
2960 IF NS=0 THEN DLTIME=10:H2CLK(W,3)=1:H2CLK(W,0)=0
2970 IF DS=0 THEN CC=2:WW=W:WSET=0:WTYP=II:WNS=NS:WLDT=LDT
2980 H2CLK(W,2)=HOUR+DLTIME:FCH=MC:IF W<>HCWAVE THEN 3000
2990 IF OCHCK=2 AND FSH<>HI THEN SSET=2:BEGSH=1 ELSE NSSET=1:BEGNSH=1
3000 GOSUB 3480:RETURN
3010 IF FMH>=HI THEN STOP
3020 MN=SNH(W,2):W=SNA(MN,1)
3030 IF DS<>0 THEN 3050 ELSE H3CLK(W,3)=1:H3CLK(W,2)=HOUR+10:FMH=SNH(W,2)
3040 GOSUB 3550:RETURN
3050 IF SNA(MN,1)<>W THEN 3110
```

```
3060 I=SNA(MN,3):IF P(I)>=SNA(MN,4) THEN 3080
3070 H3CLK(W,1)=W:H3CLK(W,2)=HOUR+10:H3CLK(W,3)=1:GOSUB 3550:RETURN
3080 P(I)=P(I)-SNA(MN,4):NCAR(I)=NCAR(I)+SNA(MN,4)
3090 IF SNA(MN,5)>LDT THEN LDT=SNA(MN,5)
3100 IF MN=NA THEN 3110 ELSE MN=MN+1:GOTO 3050
3110 H3CLK(W,1)=W:H3CLK(W,3)=2:H3CLK(W,4)=LDT
3120 FOR I=1 TO NHT:AP(I)=P(I):H3CLK(W,I+6)=NCAR(I):NEXT
3130 FOR I=1 TO NHT:IF NCAR(I)<>0 THEN II=I
3140 NEXT
3150 H3CLK(W,0)=1
3160 GOSUB 3200
3170 IF NS=0 THEN DLTIME=10:H3CLK(W,3)=1:H3CLK(W,0)=0
3180 IF DS=0 THEN CC=3:WW=W:WSET=0:WTYP=II:WNS=NS:WLDT=LDT
3189 '
3190 H3CLK(W,2)=HOUR+DLTIME:FMH=MN:GOSUB 3550:RETURN
3200 IF HOUR>1500 THEN CLOSE #2:STOP
3210 J=0:FOR I=1 TO NHT
3220 IF NCAR(I)=0 THEN 3240
3230 J=J+1:TYP=I
3240 NEXT
3250 IF J>1 THEN 3260
3260 '
3270 NS=HOST(LDT,TYP)
3280 IF NS=0 THEN RETURN
3290 IF NS>NCAR(TYP) THEN 3310
3300 FOR J=1 TO 3:FOR K=1 TO NHT:HOST(J,K)=0:NEXT:NEXT:DS=0:GOTO 3350
3310 RMFRC!=(NS-NCAR(TYP))/HOST(LDT,TYP):FOR J=1 TO 3:FOR K=1 TO NHT
3312 HOST(J,K)=RMFRC!*HOST(J,K):NEXT:NEXT
3320 DS=1
3330 'PRINT"RMFRC";RMFRC!,"NS";NS
3340 '
3350 DLTIME=0
3360 IF NCAR(TYP)<=0 THEN 3380
3370 DLTIME=DLTIME+HTF(5,LDT):NCAR(TYP)=NCAR(TYP)-NS:GOTO 3360
3380 RETURN
3390 STOP:RETURN
3399 '
3400 TIME=LATE:FOR W=1 TO HSWAVE
3410 IF H1CLK(W,2)>=TIME THEN 3430
3420 TIME=H1CLK(W,2):ACODE=H1CLK(W,3):HWAVE=W
3430 NEXT
3440 MCLOCK(1,2)=TIME:IF HWAVE>HSWAVE THEN RETURN ELSE MCLOCK(1,1)=HWAVE:MCLOCK(
1,3)=ACODE:H1CLK(HWAVE,2)=HOLD:MCLOCK(1,5)=1:MCLOCK(1,6)=1
3450 IF ACODE=6 THEN H1CLK(HWAVE,2)=HOLD
3460 RETURN
3470 '
3480 TIME=LATE:FOR W=1 TO HCWAVE
3490 IF H2CLK(W,2)>=TIME THEN 3510
3500 TIME=H2CLK(W,2):ACODE=H2CLK(W,3):HWAVE=W
3510 NEXT
3520 MCLOCK(2,2)=TIME:IF HWAVE>HCWAVE THEN RETURN ELSE MCLOCK(2,1)=HWAVE:MCLOCK(
2,3)=ACODE:H2CLK(HWAVE,2)=HOLD:MCLOCK(2,5)=1:MCLOCK(2,6)=2
3530 IF ACODE=6 THEN H2CLK(HWAVE,2)=HOLD
3540 RETURN
3549 '
3550 '
3560 TIME=LATE:FOR W=1 TO HNWAVE
3570 IF H3CLK(W,2)>=TIME THEN 3590
3580 TIME=H3CLK(W,2):ACODE=H3CLK(W,3):HWAVE=W
3590 NEXT
```

```
3600 MCLOCK(3,2)=TIME:IF HWAVE>HNWAVE THEN RETURN ELSE MCLOCK(3,1)=HWAVE:MCLOCK(
3,3)=ACODE:H3CLK(HWAVE,2)=HOLD:MCLOCK(3,5)=1:MCLOCK(3,6)=3
3610 IF ACODE=6 THEN H3CLK(HWAVE,2)=HOLD
3620 RETURN
3629 '
3630 'MSORT
3640 TIME=LATE:FOR I=1 TO 6
3650 IF MCLOCK(I,2)>=TIME THEN 3670
3660 TIME=MCLOCK(I,2):J=I
3670 NEXT
3680 IF TIME=LATE THEN END
3690 HOUR=TIME:W=MCLOCK(J,1):A=MCLOCK(J,3):C=MCLOCK(J,6):MODE=MCLOCK(J,5):PRINT"
MIN";HOUR,"CLASS";C,"WAVE";W,"ACTIVITY";A,"MODE";MODE
3694 RETURN
3699 '
3700 'HLNCH
3710 IF C>1 THEN 3810
3720 M=SSH(W,2):FOR I=1 TO NHT:IF H1CLK(W,I+6)<>0 THEN NCAR(I)=H1CLK(W,I+6):TYP=
I
3730 NEXT
3740 'PRINT"No carriers";NCAR(TYP);TYP
3750 IF SSA(M,1)<>W THEN 3770
3760 SSA(M,7)=1:IF M=TA THEN FSH=HI:GOTO 3770 ELSE M=M+1:GOTO 3750
3770 LDT=H1CLK(W,4):DT=(LZDIS+ALADIS)*60/HSPD(LDT)+HTF(6,LDT)+HTF(7,LDT)
3780 IF C=CC AND W=WW THEN WSET=1:CC=0:WW=0:DS=1:CC1=0:WW1=0
3790 GOSUB 8640
3800 H1CLK(W,2)=HOUR+DT:H1CLK(W,3)=3:GOSUB 3400:GOTO 3990
3810 IF C>2 THEN 3910
3820 MC=SCH(W,2):FOR I=1 TO NHT:IF H2CLK(W,I+6)<>0 THEN NCAR(I)=H2CLK(W,I+6):TYP
=I
3830 NEXT
3840 IF SCA(MC,1)<>W THEN 3860
3850 SCA(MC,7)=1:IF MC=CA THEN FCH=HI:GOTO 3860 ELSE MC=MC+1:GOTO 3840
3860 LDT=H2CLK(W,4):DT=(LZDIS+ALADIS)*60/HSPD(LDT)+HTF(6,LDT)+HTF(7,LDT)
3870 IF C=CC AND W=WW THEN WSET=1:CC=0:WW=0:DS=1:CC1=0:WW1=0
3880 GOSUB 8640
3890 H2CLK(W,2)=HOUR+DT:H2CLK(W,3)=3:GOSUB 3480
3900 GOTO 3990
3910 MN=SNH(W,2):FOR I=1 TO NHT:IF H3CLK(W,I+6)<>0 THEN NCAR(I)=H3CLK(W,I+6):TYP
=I
3920 NEXT
3930 IF SNA(MN,1)<>W THEN 3950
3940 SNA(MN,7)=1:IF MN=NA THEN 3950 ELSE MN=MN+1:GOTO 3930
3950 LDT=H3CLK(W,4):DT=(LZDIS+ALADIS)*60/HSPD(LDT)+HTF(6,LDT)+HTF(7,LDT)
3960 IF C=CC AND W=WW THEN WSET=1:CC=0:WW=0:DS=1:CC1=0:WW1=0
3970 GOSUB 8640
3980 H3CLK(W,2)=HOUR+DT:H3CLK(W,3)=3:GOSUB 3550
3990 PAC=HEST(LDT,TYP)-HOST(LDT,TYP)
3992 IF NCAR(TYP)>PAC THEN 4020
4000 FOR J=1 TO 3:FOR K=1 TO NHT:HOST(J,K)=HEST(J,K):NEXT:NEXT
4002 IF DS=1 THEN GOSUB 7080
4010 RETURN
4020 RMFRC!=NCAR(TYP)/PAC
4030 FOR J=1 TO 3:FOR K=1 TO NHT
4032 HOST(J,K)=HOST(J,K)+RMFRC!*(HEST(J,K)-HOST(J,K))
4034 IF HOST(J,K)>HEST(J,K) THEN HOST(J,K)=HEST(J,K)
4040 NEXT:NEXT
4042 IF DS=1 THEN GOSUB 7080
4050 RETURN
4059 '
```

```
4060 'HUNLD
4070 IF C>2 GOTO 7560 ELSE IF C>1 THEN GOTO 7370 ELSE M=SSH(W,2)
4080 W=SSA(M,1)
4090 FOR I=1 TO NHT:NCAR(I)=H1CLK(W,I+6):FOR J=1 TO 3:FOR K=1 TO NLZ
4092 DLABAT(I,J,K)=0:NEXT:NEXT:NEXT
4100 FOR K=1 TO NLZ:ZONE(K)=0:NEXT
4110 IF SSA(M,1)<>W THEN 4140 ELSE I=SSA(M,3)
4120 LDT=SSA(M,5):DEST=SSA(M,6):IF DEST<=NLZ THEN 4130 ELSE PRINT"DEST";DEST:STO
P
4130 IF ZN(DEST)>HOUR THEN H1CLK(W,2)=ZN(DEST):H1CLK(W,3)=3:GOSUB 3400:RETURN EL
SE DLABAT(I,LDT,DEST)=DLABAT(I,LDT,DEST)+SSA(M,4):ZONE(DEST)=1:IF M=TA THEN 4140
 ELSE M=M+1:GOTO 4110
4140 FOR I=1 TO NHT:IF NCAR(I)<>0 THEN L=I
4150 NEXT
4160 FOR K=1 TO NLZ:DTIME(K)=0:NEXT
4170 FOR K=1 TO NLZ:FOR J=1 TO 3:FOR I=1 TO NHT
4180 IF DLABAT(I,J,K)<=0 THEN 4190 ELSE DLABAT(I,J,K)=DLABAT(I,J,K)-HGF(I):DTIME
(K)=DTIME(K)+HTF(1,J)+HTF(2,J)+HTF(8,J):GOTO 4180
4190 NEXT:NEXT:NEXT
4200 ZTIME=0:FOR K=1 TO NLZ
4210 IF DTIME(K)>ZTIME THEN ZTIME=DTIME(K)
4220 NEXT:H1CLK(W,2)=HOUR+ZTIME:H1CLK(W,3)=4:FOR K=1 TO NLZ
4222 IF ZONE(K)=1 THEN ZN(K)=H1CLK(W,2)
4230 NEXT
4240 'PRINT "Nr of Helos";H1CLK(W,L+6)
4250 GOSUB 3400:RETURN
4260 RETURN
4269 '
4270 IF C>1 THEN 4490
4280 FOR TH=1 TO NHT
4290 IF H1CLK(W,TH+6)=0 THEN 4330
4300 NH=H1CLK(W,TH+6):IF W>=TRNH1 THEN 4320
4310 NH=1!*NH
4320 AP(TH)=AP(TH)+NH
4330 NEXT:H1CLK(W,2)=LATE
4340 'PRINT"Helo Pool";AP(1);AP(2);AP(3);AP(4)
4350 IF M=TA THEN FSH=HI
4360 IVAL=0:J=0:FOR I=1 TO HSWAVE:IF H1CLK(I,0)>0 THEN 4380
4370 J=J+1:IVAL=1:IF J=1 THEN FSH=SSH(I,2)
4380 NEXT:IF IVAL=0 THEN GOSUB 3400:GOTO 4480
4390 IF OCHCK=2 AND SSET=1 THEN GOSUB 3400:RETURN
4400 M=FSH:FSHOLD=FSH:W=SSA(M,1):IF W=0 OR H1CLK(W,0)>0 THEN 4470 ELSE SSH(W,2)=
FSH:FOR I=1 TO NHT:P(I)=AP(I):NEXT
4410 IF SSA(M,1)<>W THEN 4460
4420 I=SSA(M,3)
4430 IF P(I)>=SSA(M,4) THEN 4450 ELSE FSH=FSHOLD:GOSUB 3400
4440 RETURN
4450 P(I)=P(I)-SSA(M,4):IF M=TA THEN 4460 ELSE M=M+1:GOTO 4410
4460 H1CLK(W,1)=W:H1CLK(W,2)=HOUR:H1CLK(W,3)=1:GOSUB 3400:FSH=M:RETURN
4470 FSH=HI:GOSUB 3400
4480 RETURN
4490 IF C>2 THEN 4560
4500 FOR TH=1 TO NHT
4510 IF H2CLK(W,TH+6)=0 THEN 4530
4520 AP(TH)=AP(TH)+H2CLK(W,TH+6)
4530 NEXT:H2CLK(W,2)=LATE
4540 IF MC=CA THEN FCH=HI
4550 GOSUB 3480:RETURN
4560 FOR TH=1 TO NHT:IF H3CLK(W,TH+6)=0 THEN 4580
4570 AP(TH)=AP(TH)+H3CLK(W,TH+6)
```

```
4580 NEXT:H3CLK(W,2)=LATE
4590 IF MN=NA THEN FMH=HI
4600 ALL=1:GOSUB 3550:RETURN
4609 '
4610 DT=HTF(3,1)+HTF(4,1)+(LZDIS+ALADIS)*60/HSPD(1)
4620 IF C>1 THEN 4690
4630 M=SSH(W,2)
4640 SSA(M,7)=2:DEST=SSA(M,6):PSH(DEST)=PSH(DEST)+SSA(M,2):PASH=PASH+SSA(M,2)
4642 SFPH!(DEST)=SFPH!(DEST)+SSA(M,0):SFASH!=SFASH!+SSA(M,0):PF!=SFASH!/TWUV!
4644 IF M=TA THEN 4650 ELSE M=M+1:IF SSA(M,1)=W THEN 4640
4650 PRINT"PASH Total:";PASH:PRINT "time:";HOUR
4660 PRINT"FIREPWR ASHORE";SFASH!,PF!
4664 PRINT #2,USING "######.##";HOUR;PASH;PF!
4670 H1CLK(W,0)=2
4680 H1CLK(W,2)=DT+HOUR:H1CLK(W,3)=5:GOSUB 3400:RETURN
4690 IF C>2 THEN 4750
4700 MC=SCH(W,2)
4710 SCA(MC,7)=2:DEST=SCA(MC,6):PSH(DEST)=PSH(DEST)+SCA(MC,2):PASH=PASH+SCA(MC,2
)
4712 SFPH!(DEST)=SFPH!(DEST)+SCA(MC,0):SFASH!=SFASH!+SCA(MC,0):PF!=SFASH!/TWUV!
4714 IF MC=CA THEN 4720 ELSE MC=MC+1:IF SCA(MC,1)=W THEN 4710
4720 PRINT"PASH Total:";PASH:PRINT "time:";HOUR
4730 PRINT"FIREPWR ASHORE";SFASH!,PF!:H2CLK(W,0)=2
4734 PRINT #2,USING "######.##";HOUR;PASH;PF!
4740 H2CLK(W,2)=DT+HOUR:H2CLK(W,3)=5:GOSUB 3480:RETURN
4750 MN=SNH(W,2)
4760 SNA(MN,7)=2:DEST=SNA(MN,6):PSH(DEST)=PSH(DEST)+SNA(MN,2):PASH=PASH+SNA(MN,2
)
4762 SFASH!=SFASH!+SNA(MN,0):PF!=SFASH!/TWUV!:IF MN=NA THEN 4770 ELSE MN=MN+1:IF
 SNA(MN,1)=W THEN 4760
4770 PRINT"PASH Total:";PASH:PRINT "time:";HOUR
4780 PRINT"FIREPWR ASHORE";SFASH!,PF!:H3CLK(W,0)=2
4784 PRINT #2,USING "######.##";HOUR;PASH;PF!
4790 H3CLK(W,2)=DT+HOUR:H3CLK(W,3)=5:GOSUB 3560:RETURN
4799 '
4800 PRINT"LSSORT"
4810 TIME=LATE:FOR W=1 TO LSWAVE:IF L1CLK(W,2)>=TIME THEN 4830
4820 TIME=L1CLK(W,2):ACODE=L1CLK(W,3):LWAVE=W
4830 NEXT:IF MCLOCK(4,1)=LWAVE AND MCLOCK(4,2)=TIME AND MCLOCK(4,3)=ACODE THEN 4
850
4840 MCLOCK(4,2)=TIME:MCLOCK(4,1)=LWAVE:MCLOCK(4,3)=ACODE:L1CLK(LWAVE,2)=HOLD
4842 MCLOCK(4,5)=2:MCLOCK(4,6)=1:IF ACODE=6 THEN L1CLK(WAVE,2)=LATE
4850 RETURN
4860 '
4861 'LCSORT
4870 TIME=LATE:FOR W=1 TO LCWAVE:IF L2CLK(W,2)>=TIME THEN 4890
4880 TIME=L2CLK(W,2):ACODE=L2CLK(W,3):LWAVE=W
4890 NEXT:IF MCLOCK(5,1)=LWAVE AND MCLOCK(5,2)=TIME AND MCLOCK(5,3)=ACODE THEN 4
910
4900 MCLOCK(5,2)=TIME:MCLOCK(5,1)=LWAVE:MCLOCK(5,3)=ACODE:L2CLK(LWAVE,2)=HOLD
4902 MCLOCK(5,5)=2:MCLOCK(5,6)=2:IF ACODE=6 THEN L2CLK(LWAVE,2)=LATE
4910 RETURN
4920 '
4921 'LNSORT
4930 TIME=LATE:FOR W=1 TO LNWAVE:IF L3CLK(W,2)>=TIME THEN 4950
4940 TIME=L3CLK(W,2):ACODE=L3CLK(W,3):LWAVE=W
4950 NEXT:IF MCLOCK(6,1)=LWAVE AND MCLOCK(6,2)=TIME AND MCLOCK(6,3)=ACODE THEN 4
970
4960 MCLOCK(6,2)=TIME:MCLOCK(6,1)=LWAVE:MCLOCK(6,3)=ACODE:L3CLK(LWAVE,2)=HOLD
4962 MCLOCK(6,5)=2:MCLOCK(6,6)=3:IF ACODE=6 THEN L3CLK(LWAVE,2)=LATE
```

```
4970 RETURN
4980 '
4990 L=NLT+NBT+NCT:FOR I=1 TO L
4992 NCAR(I)=0:P(I+NHT)=AP(I+NHT):NEXT:MCAR=0:IF C>1 THEN 5120
5000 IF FSL>=HI THEN STOP
5010 LS=SSL(W,2)
5020 IF WD<>0 THEN 5040 ELSE L1CLK(W,3)=1:L1CLK(W,2)=HOLD
5030 GOSUB 4800:RETURN
5040 IF SSS(LS,1)<>W THEN 5070 ELSE I=SSS(LS,3)-NHT
5050 IF P(I+NHT)<SSS(LS,4)THEN L1CLK(W,1)=W:L1CLK(W,2)=HOUR+10:L1CLK(W,3)=1:GOSU
B 4800:RETURN
5060 P(I+NHT)=P(I+NHT)-SSS(LS,4):NCAR(I)=NCAR(I)+SSS(LS,4):MCAR=MCAR+SSS(LS,4)
5062 IF LS=TS THEN 5070 ELSE LS=LS+1:GOTO 5040
5070 L1CLK(W,1)=W:L1CLK(W,3)=2:FOR I=1 TO L
5072 AP(I+NHT)=P(I+NHT):L1CLK(W,I+6)=NCAR(I):NEXT
5080 GOSUB 5320:IF WD=0 THEN CCC=1:WWW=W:GSET=0
5090 L1CLK(W,0)=1
5100 L1CLK(W,2)=HOUR+DLTIME:FSL=LS:IF W=LSWAVE AND OCLCK=0 THEN LCSET=1:BEGOCL=1
 ELSE IF W=LSWAVE THEN NLSET=1:BEGNSL=1
5110 GOSUB 4800:RETURN
5120 IF C>2 THEN 5240 ELSE LC=SCL(W,2)
5130 IF WD<>0 THEN 5150 ELSE L2CLK(W,3)=1:L2CLK(W,2)=HOLD
5140 GOSUB 4860:RETURN
5150 IF SCS(LC,1)<>W THEN 5180 ELSE I=SCS(LC,3)-NHT
5160 IF P(I+NHT)<SCS(LC,4)THEN L2CLK(W,1)=W:L2CLK(W,2)=HOUR+10:L2CLK(W,3)=1:GOSU
B 4860:RETURN
5170 P(I+NHT)=P(I+NHT)-SCS(LC,4):NCAR(I)=NCAR(I)+SCS(LC,4):MCAR=MCAR+SCS(LC,4)
5172 IF LC=CS THEN 5180 ELSE LC=LC+1:GOTO 5150
5180 L2CLK(W,1)=W:L2CLK(W,3)=2:FOR I=1 TO L
5182 AP(I+NHT)=P(I+NHT):L2CLK(W,I+6)=NCAR(I):NEXT:GOSUB 5320:IF WD=0 THEN CCC=2:
WWW=W:GSET=0
5190 L2CLK(W.2)=HOUR+DLTIME:FCL=LC:L2CLK(W,0)=1
5200 IF W=LCWAVE AND OCLCK=2 AND LLSET=1 THEN LLSET=2:BEGSL=1
5210 IF W<>LCWAVE THEN 5230
5220 IF OCLCK=2 AND FSL<>HI THEN LLSET=2:BEGSL=1 ELSE NLSET=1:BEGNSL=1
5230 GOSUB 4860:RETURN
5240 LN=SNL(W,2)
5250 IF WD<>0 THEN 5270 ELSE L3CLK(W,3)=1:L3CLK(W,2)=HOLD
5260 GOSUB 4920:RETURN
5270 IF SNS(LN,1)<>W THEN 5300 ELSE I=SNS(LN,3)-NHT
5280 IF P(I+NHT)<SNS(LN,4)THEN L3CLK(W,1)=W:L3CLK(W,2)=HOUR+10:L3CLK(W,3)=1:GOSU
B 4920:RETURN
5290 P(I+NHT)=P(I+NHT)-SNS(LN,4):NCAR(I)=NCAR(I)+SNS(LN,4):MCAR=MCAR+SNS(LN.4)
5292 IF LN=MS THEN 5300 ELSE LN=LN+1:GOTO 5270
5300 L3CLK(W,1)=W:L3CLK(W,3)=2:FOR I=1 TO L
5302 AP(I+NHT)=P(I+NHT):L3CLK(W,I+6)=NCAR(I):NEXT:GOSUB 5320:IF WD=0 THEN CCC=3:
WWW=W:GSET=0
5310 L3CLK(W,2)=HOUR+DLTIME:FML=LN:L3CLK(W,0)=1:GOSUB 4920:RETURN
5319 '
5320 PRINT"WELDEK":NS=WDST:DLTIME=0:IF NS<=MCAR THEN WDST=0 ELSE WDST=WDST-MCAR
5330 FOR I=1 TO NLT+NBT+NCT:IF NCAR(I)<>0 THEN DTM(I)=LTF(5,I) ELSE DTM(I)=0
5340 NEXT:IF WDST=0 THEN 5370
5350 FOR I=1 TO NLT+NBT+NCT:IF DTM(I)>DLTIME THEN DLTIME=DTM(I)
5360 NEXT:RETURN
5370 WD=0:NRES=0:FOR I=1 TO NLT+NBT+NCT:NRES=NRES+NCAR(I)
5372 IF NRES<NS THEN 5380 ELSE DLTIME=DLTIME+DTM(I):NRES=NRES-NS
5380 NEXT
5390 RETURN
5399 '
```

```
5400 DT=0:MCAR=0:IF C>1 THEN 5520 ELSE LS=SSL(W,2):FOR I=1 TO NLT+NBT+NCT:NCAR(I
)=L1CLK(W,I+6):NEXT
5410 IF SSS(LS,1)=W THEN SSS(LS,7)=1:IF LS=TS THEN 5450 ELSE LS=LS+1:GOTO 5410
5420 FOR I=1 TO NLT:IF NCAR(I)<>0 THEN D=LODDIS*60/BSPD(I)+LTF(6,I)+LTF(7,I):MCA
R=MCAR+NCAR(I) ELSE D=0
5430 IF D>DT THEN DT=D
5440 NEXT
5450 FOR I=NLT+1 TO NLT+NBT+NCT
5452 IF NCAR(I)<>0 THEN D=DLADIS*60/BSPD(I)+LTF(6,I)+LTF(7,I):MCAR=MCAR+NCAR(I)
ELSE D=0
5460 IF SSL(W,1)>NHT+NLT+NBT THEN D=ASADIS*60/BSPD(I)+CLZDIS*60/CSPD+LTF(6,I)+LT
F(7,I)
5470 IF D>DT THEN DT=D
5480 NEXT
5490 IF SSL(W,1)>NHT+NLT+NBT THEN I=NLT+NBT+NCT:DT=ASADIS*60/BSPD(I)+CLZDIS*60/C
SPD+LTF(6,I)+LTF(7,I)
5500 IF C=CCC AND W=WWW THEN GSET=1:CCC=0:WWW=0:WD=1
5510 L1CLK(W,2)=HOUR+DT:L1CLK(W,3)=3:GOSUB 4800:GOTO 5760
5520 IF C>2 THEN 5640 ELSE LC=SCL(W,2):FOR I=1 TO NLT+NBT+NCT:NCAR(I)=L2CLK(W,I+
6):NEXT
5530 IF SCS(LC,1)=W THEN SCS(LC,7)=1:IF LC=CS THEN 5570 ELSE LC=LC+1:GOTO 5530
5540 FOR I=1 TO NLT:IF NCAR(I)<>0 THEN D=LODDIS*60/BSPD(I)+LTF(6,I)+LTF(7,I):MCA
R=MCAR+NCAR(I) ELSE D=0
5550 IF D>DT THEN DT=D
5560 NEXT
5570 FOR I=NLT+1 TO NLT+NBT+NCT:IF NCAR(I)<>0 THEN D=DLADIS*60/BSPD(I)+LTF(6,I)+
LTF(7,I):MCAR=MCAR+NCAR(I) ELSE D=0
5580 IF SCL(W,1)>NHT+NLT+NBT THEN D=ASADIS*60/BSPD(I)+CLZDIS*60/CSPD+LTF(6,I)+LT
F(7,I)
5590 IF D>DT THEN DT=D
5600 NEXT
5610 IF SCL(W,1)>NHT+NLT+NBT THEN I=NLT+NBT+NCT:DT=ASADIS*60/BSPD(I)+CLZDIS*60/C
SPD+LTF(6,I)+LTF(7,I)
5620 IF C=CCC AND W=WWW THEN GSET=1:CCC=0:WWW=0:WD=1
5630 L2CLK(W,2)=HOUR+DT:L2CLK(W,3)=3:GOSUB 4860:GOTO 5760
5640 LN=SNL(W,2):FOR I=1 TO NLT+NBT+NCT:NCAR(I)=L3CLK(W,I+6):NEXT
5650 IF SNS(LN,1)=W THEN SNS(LN,7)=1:IF LN=MS THEN 5690 ELSE LN=LN+1:GOTO 5650
5660 FOR I=1 TO NLT
5662 IF NCAR(I)<>0 THEN D=LODDIS*60/BSPD(I)+LTF(6,I)+LTF(7,I):MCAR=MCAR+NCAR(I)
ELSE D=0
5670 IF D>DT THEN DT=D
5680 NEXT
5690 FOR I=NLT+1 TO NLT+NBT+NCT
5692 IF NCAR(I)<>0 THEN D=DLADIS*60/BSPD(I)+LTF(6,I)+LTF(7,I):MCAR=MCAR+NCAR(I)
ELSE D=0
5700 IF D>DT THEN DT=D
5710 NEXT
5720 IF SNL(W,1)>NHT+NLT+NBT THEN I=NLT+NBT+NCT:DT=ASADIS*60/BSPD(I)+CLZDIS*60/C
SPD+LTF(6,I)+LTF(7,I)
5730 IF D>DT THEN DT=D
5740 IF C=CCC AND W=WWW THEN GSET=1:CCC=0:WWW=0:WD=1
5750 L3CLK(W,2)=HOUR+DT:L3CLK(W,3)=3:GOSUB 4920
5760 SPAC=WEST-WDST:IF MCAR<SPAC THEN WDST=WDST+MCAR ELSE WDST=WEST
5770 IF WDST>WEST THEN WDST=WEST
5780 RETURN
5789 '
5790 L=NLT+NBT+NCT
5800 ZN=NNB+NCLZ:IF C>2 THEN 6000 ELSE IF C>1 THEN 5910 ELSE LS=SSL(W,2)
5810 FOR I=1 TO L:NCAR(I)=L1CLK(W,I+6):NEXT:L1CLK(W,0)=1
5820 FOR I=1 TO L:FOR K=1 TO ZN:DBAT(I,K)=0:NEXT:NEXT
```

```
5830 FOR K=1 TO ZN:LZONE(K)=0:NEXT
5840 IF SSS(LS,1)<>W THEN 5870 ELSE I=SSS(LS,3)-NHT
5850 DEST=SSS(LS,6):IF LZN(DEST)>HOUR THEN L1CLK(W,2)=LZN(DEST):L1CLK(W,3)=3:GOS
UB 4800:RETURN
5860 DBAT(I,DEST)=DBAT(I,DEST)+SSS(LS,4):LZONE(DEST)=1
5862 IF LS=TS THEN 5870 ELSE LS=LS+1:GOTO 5840
5870 GOSUB 6090
5880 L1CLK(W,2)=HOUR+ZTIME:L1CLK(W,3)=4:FOR K=1 TO ZN
5882 IF LZONE(K)=1 THEN LZN(K)=L1CLK(W,2)
5890 NEXT:PRINT"NR OF CRAFT";L1CLK(W,LL+6)
5900 GOSUB 4800:RETURN
5910 LC=SCL(W,2)
5920 FOR I=1 TO L:NCAR(I)=L2CLK(W,I+6):FOR K=1 TO ZN
5930 IF SCS(LC,1)<>W THEN 5960 ELSE I=SCS(LC,3)-NHT
5940 DEST=SCS(LC,6):IF LZN(DEST)>HOUR THEN L2CLK(W,2)=LZN(DEST):L2CLK(W,3)=3:GOS
UB 4860:RETURN
5950 DBAT(I,DEST)=DBAT(I,DEST)+SCS(LC,4):LZONE(DEST)=1
5952 IF LC=CS THEN 5960 ELSE LC=LC+1:GOTO 5930
5960 GOSUB 6090
5970 L2CLK(W,2)=HOUR+ZTIME:L2CLK(W,3)=4:FOR K=1 TO ZN
5972 IF LZONE(K)=1 THEN LZN(K)=L2CLK(W,2)
5980 NEXT:PRINT"NR OF CRAFT";L2CLK(W,LL+6)
5990 GOSUB 4860:RETURN
5992 DBAT(I,K)=0:LZONE(K)=0:NEXT:NEXT:L2CLK(W,0)=1
6000 LN=SNL(W,2)
6010 FOR I=1 TO L:NCAR(I)=L3CLK(W,I+6):FOR K=1 TO ZN
6012 DBAT(I,K)=0:LZONE(K)=0:NEXT:NEXT:L3CLK(W,0)=1
6020 IF SNS(LN,1)<>W THEN 6050 ELSE I=SNS(LN,3)-NHT
6030 DEST=SNS(LN,6):IF LZN(DEST)>HOUR THEN L3CLK(W,2)=LZN(DEST):L3CLK(W,3)=3:GOS
UB 4920:RETURN
6040 DBAT(I,DEST)=DBAT(I,DEST)+SNS(LN,4):LZONE(DEST)=1
6042 IF LN=MS THEN 6050 ELSE LN=LN+1:GOTO 6020
6050 GOSUB 6090
6060 L3CLK(W,2)=HOUR+ZTIME:L3CLK(W,3)=4:FOR K=1 TO ZN
6062 IF LZONE(K)=1 THEN LZN(K)=L3CLK(W,2)
6070 NEXT:PRINT"NR OF CRAFT";L3CLK(W,LL+6)
6080 GOSUB 4920:RETURN
6090 FOR I=1 TO L:IF NCAR(I)<>0 THEN LL=I
6100 NEXT:FOR K=1 TO ZN:DTIME(K)=0:NEXT
6110 FOR K=1 TO ZN:FOR I=1 TO L
6120 IF DBAT(I,K)<=0 THEN 6130 ELSE DBAT(I,K)=DBAT(I,K)-LGF(I):DTIME(K)=DTIME(K)
+LTF(1,I)+LTF(2,I)+LTF(8,I):GOTO 6120
6130 NEXT:NEXT
6140 ZTIME=0:FOR K=1 TO ZN:IF DTIME(K)>ZTIME THEN ZTIME=DTIME(K)
6150 NEXT:RETURN
6159 '
6160 DT=0:ZN=NNB+NCLZ:ZIG=0:ZIGNAL=0:FOR I=1 TO NLT+NBT+NCT
6162 ZAVE(I)=0:NEXT:IF C>1 THEN 6320
6170 LS=SSL(W,2)
6180 SSS(LS,7)=2:DEST=SSS(LS,6):PSH(DEST)=PSH(DEST)+SSS(LS,2):PASH=PASH+SSS(LS,2
)
6182 SFPH!(DEST)=SFPH!(DEST)+SSS(LS,0):SFASH!=SFASH!+SSS(LS,0):PF!=SFASH!/TWUV!
6184 IF SSS(LS,3)<=NHT+NLT THEN ZIGNAL=1 ELSE ZAVE(SSS(LS,3)-NHT)=1
6190 IF SSS(LS,3)>NHT+NLT+NBT THEN ZIG=1
6200 IF LS=TS THEN 6220
6210 LS=LS+1:IF SSS(LS,1)=W THEN 6180
6220 PRINT"PASH Total:";PASH:PRINT "time:";HOUR
6230 PRINT"FIREPWR ASHORE";SFASH!,PF!
6234 PRINT #2,USING "######.##";HOUR;PASH;PF!
6240 IF ZIGNAL=1 THEN 6300
```

```
6250 IF ZIG=1 THEN I=NLT+NBT+NCT:DT=LTF(3,I)+LTF(4,I)+ASADIS*60/BSPD(I)+CLZDIS*6
0/CSPD:GOTO 6290
6260 FOR I=1 TO NLT+NBT+NCT
6262 IF ZAVE(I)=0 THEN 6280 ELSE D=LTF(3,I)+LTF(4,I)+DLADIS*60/BSPD(I)
6270 IF D>DT THEN DT=D
6280 NEXT
6290 L1CLK(W,2)=HOUR+DT:L1CLK(W,3)=5:GOTO 6310
6300 L1CLK(W,2)=LATE
6310 L1CLK(W,0)=2:GOSUB 4800:RETURN
6320 IF C>2 THEN 6480
6330 LC=SCL(W,2)
6340 SCS(LC,7)=2:PASH=PASH+SCS(LC,2):SEASH=SEASH+SCS(LC,5):SFASH!=SFASH!+SCS(LC,
0):PF!=SFASH!/TWUV!
6342 IF SCS(LC,3)<=NHT+NLT THEN ZIGNAL=1 ELSE ZAVE (SCS(LC,3)-NHT)=1
6350 IF SCS(LC,3)>NHT+NLT+NBT THEN ZIG=1
6360 IF LC=CS THEN 6380
6370 LC=LC+1:IF SCS(LC,1)=W THEN 6340
6380 PRINT"PASH Total:";PASH:PRINT "time:";HOUR
6390 PRINT"FIREPWR ASHORE";SFASH!,PF!
6394 PRINT #2,USING "######.##";HOUR;PASH;PF!
6400 IF ZIGNAL=1 THEN 6460
6410 IF ZIG=1 THEN I=NLT+NBT+NCT:DT=LTF(3,I)+LTF(4,I)+ASADIS*60/BSPD(I)+CLZDIS*6
0/CSPD:GOTO 6450
6420 FOR I=1 TO NLT+NBT+NCT:IF ZAVE(I)=0 THEN 6440 ELSE D=LTF(3,I)+LTF(4,I)+DLAD
IS*60/BSPD(I)
6430 IF D>DT THEN DT=D
6440 NEXT
6450 L2CLK(W,2)=HOUR+DT:L2CLK(W,3)=5:GOTO 6470
6460 L2CLK(W,2)=LATE
6470 L2CLK(W,0)=2:GOSUB 4860:RETURN
6480 LN=SNL(W,2)
6490 SNS(LN,7)=2:PASH=PASH+SNS(LN,2):EASH=EASH+SNS(LN,5):SFASH!=SFASH!+SNS(LN,0)
:PF!=SFASH!/TWUV!
6492 IF SNS(LN,3)<=NHT+NLT THEN ZIGNAL=1 ELSE ZAVE(SNS(LN,3)-NHT)=1
6500 IF SNS(LN,3)>NHT+NLT+NBT THEN ZIG=1
6510 IF LN=MS THEN 6530
6520 LN=LN+1:IF SNS(LN,1)=W THEN 6490
6530 PRINT"PASH Total:";PASH:PRINT "time:";HOUR:DT=0
6540 PRINT"FIREPWR ASHORE";SFASH!,PF!
6544 PRINT #2,USING "######.##";HOUR;PASH;PF!
6550 IF ZIGNAL=1 THEN 6610
6560 IF ZIG=1 THEN I=NLT+NBT+NCT:DT=LTF(3,I)+LTF(4,I)+ASADIS*60/BSPD(I)+CLZDIS*6
0/CSPD:GOTO 6600
6570 FOR I=1 TO NLT+NBT+NCT
6572 IF ZAVE(I)=0 THEN 6590 ELSE D=LTF(3,I)+LTF(4,I)+DLADIS*60/BSPD(I)
6580 IF D>DT THEN DT=D
6590 NEXT
6600 L3CLK(W,2)=HOUR+DT:L3CLK(W,3)=5:GOTO 6620
6610 L3CLK(W,2)=LATE
6620 L3CLK(W,0)=2:GOSUB 4920:RETURN
6629 '
6630 IF C>1 THEN 6810
6640 FOR TB=1 TO NLT+NBT+NCT:IF L1CLK(W,TB+6)=0 THEN 6680
6650 NB=L1CLK(W,TB+6):IF W>TRNS1 THEN 6670
6660 NB=.8*NB
6670 AP(TB+NHT)=AP(TB+NHT)+NB
6680 NEXT:L1CLK(W,2)=LATE
6682 IF NCT=0 THEN PRINT
6700 IF W=LSWAVE THEN GOSUB 4800:RETURN ELSE GOTO 6720
6710 IF SSL(W,1)>NHT+NLT+NBT THEN 6740
```

```
6720 W=W+1:IF W>LSWAVE THEN 6800
6730 IF L1CLK(W,0)=0 THEN FSL=SSL(W,2):GOTO 6710 ELSE 6720
6740 FOR I=1 TO NLT+NBT+NCT:P(I+NHT)=AP(I+NHT):NEXT
6750 IF SSS(LS,1)<>W THEN 6790
6760 I=SSS(LS,3)-NHT:IF P(I+NHT)>SSS(LS,4)THEN 6780
6770 RETURN
6780 P(I+NHT)=P(I+NHT)-SSS(LS,4):IF LS=TS THEN 6790 ELSE LS=LS+1:GOTO 6750
6790 L1CLK(W,1)=W:L1CLK(W,2)=HOUR:L1CLK(W,3)=1:GOSUB 4800:FSL=LS:RETURN
6800 FSL=HI:GOSUB 4800:RETURN
6810 IF C>2 THEN 6960
6820 FOR TB=1 TO NLT+NBT+NCT:IF L2CLK(W,TB+6)=0 THEN 6840
6830 AP(TB+NHT)=AP(TB+NHT)+L2CLK(W,TB+6)
6840 NEXT:L2CLK(W,2)=LATE
6850 IF LC=CS THEN FCL=HI
6860 IF W=LCWAVE THEN GOSUB 4860:RETURN
6870 LC=FCL:W=SCS(LC,1):IF W=0 THEN 6950 ELSE IF SCL(W,1)>NHT+NLT+NBT THEN 6890
6880 W=W+1:FCL=SCL(W,2):GOTO 6870
6890 FOR I=1 TO NLT+NBT+NCT:P(I+NHT)=AP(I+NHT):NEXT
6900 IF SCS(LC,1)<>W THEN 6940
6910 I=SCS(LC,3)-NHT:IF P(I+NHT)>=SCS(LC,4)THEN 6930
6920 RETURN
6930 P(I+NHT)=P(I+NHT)-SCS(LC,4):IF LC=CS THEN 6940 ELSE LC=LC+1:GOTO 6900
6940 L2CLK(W,1)=W:L2CLK(W,2)=HOUR:L2CLK(W,3)=1:GOSUB 4860:FCL=LC:RETURN
6950 FCL=HI:GOSUB 4860:RETURN
6960 FOR TB=1 TO NLT+NBT+NCT
6962 IF L3CLK(W,TB+6)<>0 THEN AP(TB+NHT)=AP(TB+NHT)+L3CLK(W,TB+6)
6970 NEXT:L3CLK(W,2)=LATE
6980 IF LN=MS THEN FML=HI
6990 IF W=LNWAVE THEN RETURN ELSE LN=FML:W=SNS(LN,1)
7000 IF W=0 THEN 7070 ELSE IF SNL(W,1)>NHT+NLT+NBT THEN 7020
7010 W=W+1:FML=SNL(W,2):GOTO 7000
7020 FOR I=1 TO NLT+NBT+NCT:P(I+NHT)=AP(I+NHT):NEXT
7030 IF SNS(LN,1)<>W THEN 7060
7040 I=SNS(LN,3)-NHT:IF P(I+NHT)<SNS(LN,4)THEN RETURN
7050 P(I+NHT)=P(I+NHT)-SNS(LN,4):IF LN=MS THEN 7060 ELSE LN=LN+1:GOTO 7030
7060 L3CLK(W,1)=W:L3CLK(W,2)=HOUR:L3CLK(W,3)=1:GOSUB 4920:FML=LN:RETURN
7070 FML=HI:GOSUB 4920:RETURN
7079 '
7080 WSET=0:J=1
7090 CVAL=0:FOR I=1 TO HSWAVE:IF H1CLK(I,3)>1 THEN 7120
7100 H1CLK(I,3)=1:H1CLK(I,2)=HOUR+J:IF J=1 THEN K=I
7110 J=J+5:CVAL=1
7120 NEXT:IF K>HSWAVE OR CVAL=0 THEN 7170
7130 IF MCLOCK(1,3)=1 OR MCLOCK(1,3)=6 THEN 7160
7140 IF H1CLK(K,2)>MCLOCK(1,2)THEN 7170
7150 HWAVE=MCLOCK(1,1):H1CLK(HWAVE,2)=MCLOCK(1,2)
7152 'PRINT"Sched Wave";HWAVE;" Time";H1CLK(HWAVE,2);" Code";H1CLK(HWAVE,3)
7160 GOSUB 3400
7170 WSET=0:IF FCH=HI THEN 7280
7180 IF CVAL=1 AND OCHCK=0 THEN RETURN
7190 IF OCHCK=2 AND SSET=0 THEN 7280
7200 CVAL=0:FOR I=1 TO HCWAVE:IF H2CLK(I,3)>1 THEN 7230
7210 H2CLK(I,3)=1:H2CLK(I,2)=HOUR+J:IF J=1 THEN K=I
7220 J=J+5:CVAL=1
7230 NEXT:IF K>HCWAVE OR CVAL=0 THEN 7280
7240 IF MCLOCK(2,3)=1 THEN 7270
7250 IF H2CLK(K,2)>MCLOCK(2,2)THEN 7280
7260 HWAVE=MCLOCK(2,1):H2CLK(HWAVE,2)=MCLOCK(2,2)
7262 'PRINT"On-Call Wave";HWAVE;" Time";H2CLK(HWAVE,2);" Code";H2CLK(HWAVE,3)
7270 GOSUB 3480
```

```
7280 WSET=0:IF FMH=HI OR HNWAVE=0 THEN RETURN
7290 CVAL=0:FOR I=1 TO HNWAVE:IF H3CLK(I,3)>1 THEN 7320
7300 H3CLK(I,3)=1:H3CLK(I,2)=HOUR+J:IF J=1 THEN K=I
7310 J=J+5:CVAL=1
7320 NEXT:IF K>HNWAVE OR CVAL=0 THEN RETURN
7330 IF MCLOCK(3,3)=1 THEN 7360
7340 IF H3CLK(K,2)>MCLOCK(3,2)THEN RETURN
7350 HWAVE=MCLOCK(3,1):H3CLK(HWAVE,2)=MCLOCK(3,2)
7360 GOSUB 3550:RETURN
7370 MC=SCH(W,2)
7380 W=SCA(MC,1)
7390 FOR I=1 TO NHT:NCAR(I)=H2CLK(W,I+6):FOR J=1 TO 3:FOR K=1 TO NLZ
7392 DLABAT(I,J,K)=0:NEXT:NEXT:NEXT
7400 FOR K=1 TO NLZ:ZONE(K)=0:NEXT
7410 IF SCA(MC,1)<>W THEN 7430 ELSE I=SCA(MC,3)
7420 LDT=SCA(MC,5):DEST=SCA(MC,6):IF ZN(DEST)>HOUR THEN H2CLK(W,2)=ZN(DEST):H2CL
K(W,3)=3:GOSUB 3480:RETURN ELSE DLABAT(I,LDT,DEST)=DLABAT(I,LDT,DEST)+SCA(MC,4):
ZONE(DEST)=1:IF MC=CA THEN 7430 ELSE MC=MC+1:GOTO 7410
7430 FOR I=1 TO NHT:IF NCAR(I)<>0 THEN L=I
7440 NEXT
7450 FOR K=1 TO NLZ:DTIME(K)=0:NEXT
7460 FOR K=1 TO NLZ:FOR J=1 TO 3:FOR I=1 TO NHT
7470 IF DLABAT(I,J,K)<=0 THEN 7480 ELSE DLABAT(I,J,K)=DLABAT(I,J,K)-HGF(I):DTIME
(K)=DTIME(K)+HTF(1,J)+HTF(2,J)+HTF(8,J):GOTO 7470
7480 NEXT:NEXT:NEXT
7490 ZTIME=0:FOR K=1 TO NLZ
7500 IF DTIME(K)>ZTIME THEN ZTIME=DTIME(K)
7510 NEXT:H2CLK(W,2)=HOUR+ZTIME:H2CLK(W,3)=4:FOR K=1 TO NLZ
7512 IF ZONE(K)=1 THEN ZN(K)=H2CLK(W,2)
7520 NEXT
7530 'PRINT "Nr of Helos";H2CLK(W,L+6)
7540 GOSUB 3480:RETURN
7550 RETURN
7560 MN=SNH(W,2)
7570 W=SNA(MN,1)
7580 FOR I=1 TO NHT:NCAR(I)=H3CLK(W,I+6):FOR J=1 TO 3:FOR K=1 TO NLZ
7582 DLABAT(I,J,K)=0:NEXT:NEXT:NEXT
7590 FOR K=1 TO NLZ:ZONE(K)=0:NEXT
7600 IF SNA(MN,1)<>W THEN 7620 ELSE I=SNA(MN,3)
7610 LDT=SNA(MN,5):DEST=SNA(MN,6):IF ZN(DEST)>HOUR THEN H3CLK(W,2)=ZN(DEST):H3CL
K(W,3)=3:GOSUB 3550:RETURN ELSE DLABAT(I,LDT,DEST)=DLABAT(I,LDT,DEST)+SNA(MN,4):
ZONE(DEST)=1:IF MN=NA THEN 7620 ELSE MN=MN+1:GOTO 7600
7620 FOR I=1 TO NHT:IF NCAR(I)<>0 THEN L=I
7630 NEXT
7640 FOR K=1 TO NLZ:DTIME(K)=0:NEXT
7650 FOR K=1 TO NLZ:FOR J=1 TO 3:FOR I=1 TO NHT
7660 IF DLABAT(I,J,K)<=0 THEN 7670 ELSE DLABAT(I,J,K)=DLABAT(I,J,K)-HGF(I):DTIME
(K)=DTIME(K)+HTF(1,J)+HTF(2,J)+HTF(8,J):GOTO 7660
7670 NEXT:NEXT:NEXT
7680 ZTIME=0:FOR K=1 TO NLZ
7690 IF DTIME(K)>ZTIME THEN ZTIME=DTIME(K)
7700 NEXT:H3CLK(W,2)=HOUR+ZTIME:H3CLK(W,3)=4:FOR K=1 TO NLZ
7702 IF ZONE(K)=1 THEN ZN(K)=H3CLK(W,2)
7710 NEXT
7720 'PRINT "Nr of Helos":H3CLK(W,L+6)
7730 GOSUB 3550:RETURN
7740 RETURN
7749 '
7750 PRINT"STOCH":BEGOCH=0:IF OCHCK=2 THEN RETURN
7760 IF OCHCK=0 THEN FCH=1:GOTO 7790
```

```
7770 FOR I=1 TO HCWAVE:IF H2CLK(I,3)=6 THEN H2CLK(I,3)=1:H2CLK(I,1)=I
7780 NEXT:GOTO 7810
7790 OCSET=0:FOR I=1 TO HCWAVE:IF H2CLK(I,0)=0 THEN H2CLK(I,1)=I:H2CLK(I,2)=HOLD
:H2CLK(I,3)=1
7800 NEXT
7810 GOSUB 7080:RETURN
7819 '
7820 PRINT"STNSH":FMH=1:BEGNSH=0:NSSET=0
7830 FOR I=1 TO HNWAVE:H3CLK(I,1)=I:IF H3CLK(I,0)=0 THEN H3CLK(I,2)=HOLD:H3CLK(I
,3)=1
7840 NEXT
7850 GOSUB 7080:RETURN
7859 '
7860 PRINT"STSH":BEGSH=0
7870 FOR I=1 TO HSWAVE
7872 IF H1CLK(I,0)=0 THEN H1CLK(I,1)=I:H1CLK(I,2)=HOLD:H1CLK(I,3)=1:PRINT"H1CLK(
I,1)";H1CLK(I,1),"H1CLK(I,3)";H1CLK(I,3)
7880 NEXT
7890 GOSUB 7080:RETURN
7899 '
7900 PRINT"HALTSH":SSET=1
7910 FOR I=1 TO HSWAVE
7912 IF H1CLK(I,0)=0 THEN H1CLK(I,1)=I:H1CLK(I,2)=HOLD:H1CLK(I,3)=6:PRINT"H1CLK(
I,1)";H1CLK(I,1),"H1CLK(I,3)";H1CLK(I,3)
7920 NEXT:RETURN
7929 '
7930 PRINT"BESTART":GSET=0:J=1:IF FSL=HI THEN 8020
7940 CVAL=0:FOR I=1 TO LSWAVE:IF L1CLK(I,3)<>1 THEN 7970
7950 L1CLK(I,2)=HOUR+J:IF J=1 THEN K=I
7960 J=J+5:CVAL=1
7970 NEXT:IF K>LSWAVE OR CVAL=0 THEN 8020
7980 IF MCLOCK(4,3)=1 OR MCLOCK(4,3)=6 THEN 8010
7990 IF L1CLK(K,2)>MCLOCK(4,2)THEN 8020
8000 LWAVE=MCLOCK(4,1):L1CLK(LWAVE,2)=MCLOCK(4,2)
8002 PRINT"Sched Wave";LWAVE;" Time";L1CLK(LWAVE,2);" Code";L1CLK(LWAVE,3)
8010 GOSUB 4800
8020 GSET=0:IF FCL=HI THEN 8120
8030 IF OCLCK=2 AND LLSET=0 THEN 8120
8040 CVAL=0:FOR I=1 TO LCWAVE:IF L2CLK(I,3)<>1 THEN 8070
8050 L2CLK(I,2)=HOUR+J:IF J=1 THEN K=I
8060 J=J+5:CVAL=1
8070 NEXT:IF K>LCWAVE OR CVAL=0 THEN 8120
8080 IF MCLOCK(5,3)=1 THEN 8110
8090 IF L2CLK(K,2)>MCLOCK(5,2)THEN 8120
8100 LWAVE=MCLOCK(5,1):L2CLK(LWAVE,2)=MCLOCK(5,2)
8102 PRINT"On-Call Wave";LWAVE;" Time";L2CLK(LWAVE,2);" Code";L2CLK(LWAVE,3)
8110 GOSUB 4860
8120 GSET=0:IF FML=HI OR LNWAVE=0 THEN RETURN
8130 CVAL=0:FOR I=1 TO LNWAVE:IF L3CLK(I,3)<>1 THEN 8160
8140 L3CLK(I,2)=HOUR+J:IF J=1 THEN K=I
8150 J=J+5:CVAL=1
8160 NEXT:IF K>LNWAVE OR CVAL=0 THEN RETURN
8170 IF MCLOCK(6,3)=1 THEN 8200
8180 IF L3CLK(K,2)>MCLOCK(6,2)THEN RETURN
8190 LWAVE=MCLOCK(6,1):L3CLK(LWAVE,2)=MCLOCK(6,2)
8200 GOSUB 4920:RETURN
8210 PRINT"STOCL":BEGOCL=0:IF OCLCK=2 THEN RETURN
8220 IF OCLCK=0 THEN FCL=1:GOTO 8250
8230 FOR I=1 TO LCWAVE:IF L2CLK(I,3)=6 THEN L2CLK(I,3)=1:L2CLK(I,1)=I
8240 NEXT:GOTO 8270
```

```
8250 LCSET=0:FOR I=1 TO LCWAVE:IF L2CLK(I,0)=0 THEN L2CLK(I.1)=I:L2CLK(I.2)=HOLD
:L2CLK(I,3)=1
8260 NEXT
8270 GOSUB 7930:RETURN
8279 '
8280 PRINT"STNSL":FML=1:BEGNSL=0:NLSET=0
8290 FOR I=1 TO LNWAVE:L3CLK(I,1)=I:IF L3CLK(I,0)=0 THEN L3CLK(I,2)=HOLD:L3CLK(I
,3)=1
8300 NEXT
8310 GOSUB 7930:RETURN
8319 '
8320 PRINT"STSL":BEGSL=0
8330 FOR I=1 TO LSWAVE
8332 IF L1CLK(I,0)=0 THEN L1CLK(I,1)=I:L1CLK(I.2)=HOLD:L1CLK(I,3)=1:PRINT"L1CLK(
I,1)";L1CLK(I,1),"L1CLK(I,3)";L1CLK(I,3)
8340 NEXT
8350 GOSUB 7930:RETURN
8359 '
8360 PRINT"HALTSL":LLSET=1
8370 FOR I=1 TO LSWAVE
8372 IF L1CLK(I,0)=0 THEN L1CLK(I,1)=I:L1CLK(I,2)=HOLD:L1CLK(I,3)=6:PRINT"L1CLK(
I,1)";L1CLK(I,1),"L1CLK(I.3)";L1CLK(I,3)
8380 NEXT:RETURN
8389 '
8390 FOR J=1 TO 3:FOR K=1 TO NHT:WHOST(J,K)=HOST(J,K):NEXT:NEXT
8400 IF CC1>1 THEN 8480
8410 IF H1CLK(WW1,2)=HOLD THEN RETURN
8420 HACLK=H1CLK(WW1,2):NCAR(WTYP)=H1CLK(WW1,WTYP+6):GOSUB 8710
8430 IF NS=0 THEN DLTIME=100
8440 HBCLK=HOUR+DLTIME:IF HACLK<=HBCLK THEN WDS=1:GOTO 8620
8450 FOR J=1 TO 3:FOR K=1 TO NHT
8452 HOST(J,K)=WHOST(J,K):NEXT:NEXT:H1CLK(WW1,2)=HBCLK
8460 PRINT HACLK:HBCLK:IF MCLOCK(1,1)=WW1 THEN MCLOCK(1,2)=HBCLK
8470 GOTO 8620
8480 IF CC1>2 THEN 8560
8490 IF H2CLK(WW1,2)=HOLD THEN RETURN
8500 HACLK=H2CLK(WW1,2):NCAR(WTYP)=H2CLK(WW1.WTYP+6):GOSUB 8710
8510 IF NS=0 THEN DLTIME=100
8520 HBCLK=HOUR+DLTIME:IF HACLK<=HBCLK THEN WDS=1:GOTO 8620
8530 FOR J=1 TO 3:FOR K=1 TO NHT
8532 HOST(J,K)=WHOST(J,K):NEXT:NEXT:H2CLK(WW1,2)=HBCLK
8540 IF MCLOCK(2.1)=WW1 THEN MCLOCK(2,2)=HBCLK
8550 GOTO 8620
8560 IF H3CLK(WW1,2)=HOLD THEN RETURN
8570 HACLK=H3CLK(WW1,2):NCAR(WTYP)=H3CLK(WW1,WTYP+6):GOSUB 8710
8580 IF NS=0 THEN DLTIME=100
8590 HBCLK=HOUR+DLTIME:IF HACLK<=HBCLK THEN WDS=1:GOTO 8620
8600 FOR J=1 TO 3:FOR K=1 TO NHT
8602 HOST(J,K)=WHOST(J,K):NEXT:NEXT:H3CLK(WW1,2)=HBCLK
8610 IF MCLOCK(3,1)=WW1 THEN MCLOCK(3,2)=HBCLK
8620 IF WDS=0 THEN DS=0:CC=CC1:WW=WW1 ELSE CC=0:WW=0:DS=1:WSET=1
8630 CC1=0:WW1=0:WNS=0:WTYP=0:WLDT=0:RETURN
8639 '
8640 IF CC=0 THEN RETURN ELSE CC1=0:WW1=0
8650 IF CC<>1 THEN 8670
8660 IF H1CLK(WW,3)<>2 OR H1CLK(WW,0)<>1 THEN RETURN ELSE 8700
8670 IF CC<>2 THEN 8690
8680 IF H2CLK(WW,3)<>2 OR H2CLK(WW,0)<>1 THEN RETURN ELSE 8700
8690 IF H3CLK(WW,3)<>2 OR H3CLK(WW,0)<>1 THEN RETURN
8700 CC1=CC:WW1=WW:RETURN
```

```
LIST 8701-8860
8709 '
8710 WPAC=HEST(WLDT,WTYP)-WHOST(WLDT,WTYP)
8712 IF WNS<WPAC THEN 8730
8720 FOR J=1 TO 3:FOR K=1 TO NHT:WHOST(J,K)=HEST(J,K):NEXT:NEXT:GOTO 8770
8730 RMFRC!=WNS/WPAC
8740 FOR J=1 TO 3:FOR K=1 TO NHT
8742 WHOST(J,K)=WHOST(J,K)+RMFRC!*(HEST(J,K)-WHOST(J,K))
8750 IF WHOST(J,K)>HEST(J,K)THEN WHOST(J,K)=HEST(J,K)
8760 NEXT:NEXT
8770 NS=WHOST(WLDT,WTYP)
8780 IF NS>NCAR(WTYP)THEN 8800
8790 FOR J=1 TO 3:FOR K=1 TO NHT:WHOST(J,K)=0:NEXT:NEXT:WDS=0:GOTO 8810
8800 RMFRC!=(NS-NCAR(WTYP))/NS:FOR J=1 TO 3:FOR K=1 TO NHT
8802 WHOST(J,K)=RMFRC!*WHOST(J,K):NEXT:NEXT:WDS=1
8810 DLTIME=0
8820 IF NS=0 THEN RETURN
8830 IF NCAR(WTYP)<=0 THEN 8860
8840 DLTIME=DLTIME+HTF(5,WLDT):NCAR(WTYP)=NCAR(WTYP)-NS:GOTO 8830
8850 PRINT WDS;CC1;WW1;CC;WW;WSET;WNS:RETURN
8860 GOTO 8850
Ok
```

LIST OF REFERENCES

1. Brewer, G.D. and Shubick, M., *The War Game*, Harvard University Press, 1979.

2. Szmczak, W.S., *Transferability of Combat Models: Limitations Imposed by Documentaion Practices*, M.S. Thesis, Naval Postgraduate School, 1979.

3. Shannon, R.E., *Systems Simulation the Art and Science*, pp.21-32, Prentice-Hall, Inc., 1975.

4. Emshoff, J.R. and Sisson, R.L., *Design and Use of Computer Simulation Models*, pp.50-59, The Macmillan Company, 1970.

5. Department of Defense Publication, *Doctrine For Amphibious Operations, August, 1967*.

6. US Marine Corps Publucation, *Doctrine For Landing Forces*, August 1971.

7. US Marine Corps FMFM 5-3, *Assault Support*, 3 May 1979.

8. Taylor, J.G., *Lanchester Models of Warfare*, p.631, ORSA, 1983.

9. U.S. Marine Corps Publication, *Amphibious Lift Factors Study (1983 and 1994), Final Report*, pp. 22-50, Plans and Evaluations Branch, HQMC, January 1984.

10. Weiss, E.H., "Usability: Toward a Science of User Documentation," *Computerworld*, v.XVII, No.2, pp.8-16, 28 February 1983.

11. National Bureau of Standards Special Publication 500-59, *Computer Model Documentation: A Review and an Approach*, by S.I. Gass, February 1979.

12. Fried, L., "Manual Must Give User, Analyst Views," *Computerworld*, v.XVI, No.35, pp.39-40, 30 August 1982 and v.XVI, No.36, pp.47-48, 6 September 1892.

13. National Bureau of Standards Special Publication 500-73, *Computer Model Documentation Guide*, Federal Computer Performance Evaluation and Simulation Center, January 1981.

BIBLIOGRAPHY

Barton, R.F., *Simulation and Gaming, a Primer*, Prentice-Hall,Inc., 1975.

Bonder, S., "Systems Analysis, A Purely Intellectual Activity," *Military Review* *51*,No.2, 1971.

Clark, F., "Software Documentation a Major Problem," *Computerworld* , v.XVII, No.9, 28 February 1983.

Keen, P.G.W., *Decision Support Systems-An Organizational Perspective*, Addison-Wesley, 1978.

US Marine Corps Publication IP 3-10, *The Landing Plan*, June 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Adjunct Professor James J. Wayman, Code 53 Department of Mathematics Naval Postgraduate School Monterey, California 93943	4
4.	Analysis Support Branch (Attn: Major S. M. Ritacco) Development Center Marine Corps Development and Education Command Quantico, Virginia 22134	15

135375
8

210774

Thesis
R5786 Ritacco
c.1 An amphibious ship-to-shore simulation for use on an IBM PC.

1 MAY 86 31441
24 AUG 88 32985
16 FEB 91 35786

210774

Thesis
R5786 Ritacco
c.1 An amphibious ship-to-shore simulation for use on an IBM PC.

thesR5786
An amphibious ship-to-shore simulation f
3 2768 001 91369 2
DUDLEY KNOX LIBRARY